BOLT BERANEK AND NEWMAN INC

CONSULTING • DEVELOPMENT • RESEARCH

A CULTURE OF INNOVATION

INSIDER ACCOUNTS OF COMPUTING AND LIFE AT BBN

A SIXTY YEAR REPORT

18 October 1948 to 1 July 2010

Submitted to:

Employees, previous employees, historians, and others
Worldwide

This research was supported by the volunteer labor of many BBNers past and present, and this is not a Raytheon BBN publication

CAMBRIDGE NEW YORK CHICAGO LOS ANGELES SAN FRANCISCO

A Culture of Innovation

About the front cover design

For decades BBN used a design for reports to customers that consisted of grayish cover paper, black lettering, and a GBC binding. For sentimental reasons, our cover for this book mimics that cover design. We do this with permission of Raytheon BBN Technologies. However, **despite the look of this book's cover, the book is not a Raytheon BBN publication and Raytheon BBN Technologies Corp. had no editorial control over its content.**

We mimic the particular details of the cover design of BBN's ARPA Computer Network Quarterly Technical Report No. 15 to ARPA, covering the interval from 1 July 1972 to 30 September 1972. At that time BBN acoustics activities worked out of the offices in the other cities as well from Cambridge.

A Culture of Innovation

Insider Accounts of Computing and Life at BBN

David Walden and Raymond Nickerson, editors

with chapters by

Nineteen Long-time BBN People

Waterside Publishing

Waterside Publishing
East Sandwich, MA
www.walden-family.com/bbn

ISBN978-0-9789737-0-4
Second printing, November 2012

To the people of BBN, whose intelligence, curiosity, and determination over the years created the work environment and computing innovations that make the BBN story worth telling.

Contents

Preface

Our purpose

Bolt Beranek and Newman (BBN) was originally a partnership and then a public corporation, Bolt Beranek and Newman Inc. As a public company, BBN went through several other organizational and name transitions. Starting in the 1990s, it became a part of a couple of big telephone companies; and then (2003–2009) it operated as BBN Technologies, a privately held corporation. Today (2011) it operates as Raytheon BBN Technologies. Throughout these incarnations, BBN has been a notable (we might claim renowned) science and engineering innovator in, first, the acoustics field and, later, the computer field. BBN's role in the development of the Internet may be its most widely known innovative involvement, but it has made equally important contributions to other, less widely known, areas of the application of computers.

This book covers BBN's history of work in the computer field,[1] as well as more general discussion of BBN's culture and management, told by people who were deeply involved in these activities for many years (some to the present day). Thus, we have titled this book *A Culture of Innovation: Insider Accounts of Computing and Life at BBN*.

The raw material for the book was originally pulled together mostly in the early- to mid-2000s, covering the period up to the early 1990s. Some, but not a lot, of more recent history has been added. Thus, *the coverage in this book of BBN's computer history is increasingly thin for the years moving forward from the mid-1990s.*

Organization and style of this volume

As can be seen from the Table of Contents, this volume is divided into several logical sections: one that is more about company history, one that is more about business and culture, one that is focused on a variety of areas of computer application, and one that focuses on the development of computer technology itself.

We mostly have attempted to use a consistent style throughout this book. However, for practical reasons of reducing editorial and keyboarding work, we have not forced a standard footnote and endnote style or bibliographic citation style on the separate chapters. For bibliographic entries for BBN reports, we also have taken a shortcut and left out the full company name and the location; the BBN library in Cambridge, Massachusetts, maintains the archive of BBN reports.

NB: While most of the chapters are told in the voices of their author or authors, Chapters 4, 16, 19, 20, and 21 were compiled by Walden and are based on extensive use of quotations from actual participants. It might have been stylistically better (and perhaps more readable) if Walden had written these chapters in his own words based on the history he learned from the quoted individuals; however, Walden judged it more

[1] A significant part of BBN's acoustic history is reported in Deborah Melone and Eric W. Wood's 2005 book *Sound Ideas — Acoustical Consulting at BBN and Acentech* (Acentech Incorporated, 33 Moulton Street, Cambridge, MA 02138). The acoustic history of BBN is also covered to some extent in Leo Beranek's 2008 memoir, *Riding the Waves: A Life in sound, Science, and Industry* (MIT Press, Cambridge, MA).

important to make available the quotes of the people who were there and did the work rather than writing his own version of the history.

Website

We have created a website to go with this book:

`www.walden-family.com/bbn`

Posted on the website are color versions of some of the book's figures that show up better in color — from Chapters 8, 11, and 13.

Corrections and additional content will also be posted on the website.

Acknowledgments

As editors of this volume, we have many people to thank. First and foremost, thanks must go to our authors. Current BBN president Tad Elmer gave us access to BBN's archives. BBN librarian Jennie Connolly and her co-workers helped us in numerous ways, as did many other BBNers and ex-BBNers. We have dedicated this book to the people of BBN whose intelligence, curiosity, and determination over the years created the computing innovations and work environment that make the BBN story worth telling. Naturally we have in mind the managers, scientists, engineers and consultants; we also are thinking of the people in the numerous technical, administrative and other support functions that an organization like BBN requires.

Investigating the history of computing at BBN and pulling together the content of this volume was prompted by an invitation from Thomas (Tim) Bergin, then editor-in-chief of the *IEEE Annals of the History of Computing*, who invited us to edit a special issue of that journal on computing at BBN. We are grateful to Tim for his invitation and for shepherding that project through the production of approximately twenty draft papers. Later, David Grier became editor-in-chief of the *IEEE Annals of the History of Computing*, and he worked with us to shape twelve of our draft documents into finished papers, suitable for publication in two special issues of the journal (volume 27 number 2 April–June 2005 and volume 28 number 1 January–March 2006). Also supporting publication of those papers were the anonymous reviewers and IEEE Computer Society employees Robin Baldwin, Alkenia Winston, Louise O'Donald, and others whose names we never knew.

We are grateful to the IEEE Computer Society for permission to reuse content from those special issues, as follows:

- A shorter version of the chapter by Leo Beranek, "BBN's Earliest Days: Founding a Culture of Engineering Creativity," appeared in the *IEEE Annals of the History of Computing*, Volume 27, Number 2, April–June 2005, pp. 6–14.
- A shorter and somewhat different version of the chapter by John Swets, "The ABCs of BBN," appeared in the *IEEE Annals of the History of Computing*, Volume 27, Number 2, April–June 2005, pp. 15–29.
- A shorter version of the chapter by Stephen Levy, "History of Technology Transfer at BBN," appeared in the *IEEE Annals of the History of Computing*, Volume 27, Number 2, April–June 2005, pp. 30–38.
- A similar version of the chapter by Frank Heart, "Leading at Top-Notch R&D Group in the BBN Environment," appeared in the *IEEE Annals of the History of Computing*, Volume 27, Number 2, April–June 2005, pp. 39–51.

- A shorter and somewhat different version of the chapter by Sheldon Baron, "Control Systems R&D at BBN," appeared in the *IEEE Annals of the History of Computing*, Volume 27, Number 2, April–June 2005, pp. 52–64.
- A shorter version of the chapter by Richard Estrada and Edward Starr, "50 Years of Acoustic Signal Processing for Detection: Coping with the Digital Revolution," appeared in the *IEEE Annals of the History of Computing*, Volume 27, Number 2, April–June 2005, pp. 65–78.
- A similar version of the chapter by Paul Castleman, "Medical Applications of Computers at BBN," appeared in the *IEEE Annals of the History of Computing*, Volume 28, Number 1, January–March 2006, pp. 6–16.
- A much shorter version of the chapter by Wallace Feurzeig, "Educational Technology at BBN," appeared in the *IEEE Annals of the History of Computing*, Volume 28, Number 1, January–March 2006, pp. 18–31.
- A shorter version of the chapter by John Makhoul, "Speech Processing at BBN," appeared in the *IEEE Annals of the History of Computing*, Volume 28, Number 1, January–March 2006, pp. 32–45.
- A shorter version of the chapter by Ralph Weischedel, "Natural-Language Understanding at BBN," appeared in the *IEEE Annals of the History of Computing*, Volume 28, Number 1, January–March 2006, pp. 46–55.
- A somewhat different version of the chapter by Steven Blumenthal, Alexander McKenzie, Craig Partridge, and David Walden, "Data Networking @ BBN," appeared in the *IEEE Annals of the History of Computing*, Volume 28, Number 1, January–March 2006, pp. 56–71.
- A shorter and somewhat different version of the chapter by Richard Schantz, "BBN's Network Computing Software Infrastructure and Distributed Applications (1970–1990)," appeared in the *IEEE Annals of the History of Computing*, Volume 28, Number 1, January–March 2006, pp. 72–88.

Several previously unpublished chapters are also included, five of which benefitted from the review process of the *IEEE Annals* although the papers were eventually withdrawn from consideration for publication in the *Annals* because of space limitations in the two BBN special issues.

Karl Berry and Steve Peter provided guidance for the use of LaTeX for typesetting this book, and Steve Peter provided font advice for the cover art and for denoting sidebars. Boris Veytsman helped with a key typesetting issue. Ulrike Fischer providing a helpful answer to a question posted to the `comp.text.tex` discussion group. Other members of the TeX community provided other answers. Jay Howland provided editorial help to Dave Walden on chapters for which he was involved in the writing or compilation. All our authors reviewed the proof copies of their own chapters, while Alex McKenzie spotted errors in the proof copies of his chapter and all of the other chapters.

Our spouses have our great appreciation for putting up with, as they have done so often in the past, the long hours we have spent at the keyboard writing and editing.

David Walden, East Sandwich, Massachusetts
Raymond Nickerson, Bedford, Massachusetts
December 2011

About the Authors

All of the authors in this volume had or are having long careers at BBN. Among them they cover the entire history of BBN.

Beranek and Levy each served as chief executive officer of the company for a long period of time. Beranek, Swets, Levy, Walden, Heart, and Starr served at various times as leader of BBN's research, development, and consulting business, where most of the activity described in this volume took place. The other authors all held or still hold senior technical or management positions in the company.

Shelly Baron joined BBN in 1967 after a 10-year career with NASA and remained there until his retirement as a senior vice president in 1998. His technical contributions were varied and well-recognized; he also held a number of management positions in his tenure at BBN. Baron received a PhD in applied mathematics from Harvard University. A Life Fellow of the IEEE, he was secretary/treasurer of the IEEE Control Systems Society, a member of the administrative committee of the IEEE Systems, Man and Cybernetics Society. In 1984, Baron was a recipient of the IEEE Centennial Medal.

Leo Beranek, who cofounded BBN in 1948, was BBN's president and CEO from 1953 to 1969 and chief scientist from 1969 till mid-1971. Previously, he was an associate professor of electrical engineering at MIT (1947–1958) and a faculty instructor of physics and communication engineering at Harvard University (1943–1947). Beranek, a Fellow of the IEEE, served on the IRE Committee on Professional Groups (1947–1948) and was Charter Chairman of the first group, Professional Group on Audio, now the IEEE Signal Processing Society. He earned a DSc from Harvard in acoustics and is the recipient of numerous awards, including the 2003 U.S. Presidential National Medal of Science.

Steve Blumenthal is the CTO at BridgePort Networks, a venture-backed start-up developing systems to provide seamless roaming between wireless LANs and cellular carrier networks. Blumenthal was with BBN from 1977 through 1997, where he worked on and led DARPA projects in packet voice/video conferencing, satellite packet switching, and the engineering and buildout of IP networks for AOL and BBN Planet. Later, at GTE and Genuity, he led the engineering of GTE's nationwide fiber-optic backbone and the development of Internet services. He has a BS and an MS in electrical engineering and computer science from MIT.

Paul Castleman joined BBN in 1962 while finishing his AB degree in applied mathematics at Harvard College. For more than two decades he directed BBN's medical computer activities. Castleman then went on to help start — and serve as chairman of the board for — Belmont Research and, later, Lincoln Technologies, two medical/pharmaceutical software firms that were each eventually successfully sold.

Dick Estrada joined BBN in 1975 after working at Bell Labs for six years. He worked on acoustic signal and information processing for 20 years. He held numerous technical and management positions at BBN and is now a part-time consultant with the company. Estrada has a PhD from the University of California at Berkeley.

Wally Feurzeig, BBN Principal Scientist, is a mathematician and computer scientist who has worked in computer science research since 1950. The central focus of his work is the development of sophisticated systems for learning and teaching in the areas of artificial intelligence, programming languages, and mathematics education. He holds an MS in mathematics from the Illinois Institute of Technology.

Sandy Fidell joined BBN shortly after receiving a PhD in experimental psychology from the University of Michigan in 1968. He was a member of the Psychoacoustics Department in Los Angeles (Van Nuys and Canoga Park), California for 33 years. For most of that time he was responsible for the computer system that supported the entire Los Angeles group and for his last 20 years at BBN was the manager of the Psychoacoustics Department. Throughout his tenure, he did research on numerous aspects of psychoacoustics. He left BBN in 2001 to found Fidell Associates, where he still serves as CEO.

Tom Fortmann received a BS in Physics from Stanford and a PhD in Electrical Engineering from MIT. After teaching at Newcastle University in Australia, he spent 1974–1997 at BBN Laboratories as an engineer, manager, and senior vice president. Since his retirement, he has taught mathematics as a volunteer in two Boston high schools, founded a math professional development program for elementary teachers, and served on the Massachusetts Board of Elementary and Secondary Education. He and his wife Carla live in a 215-year-old house on the Battle Green in Lexington, where they raise chickens and Navy SEALs.

Frank Heart, after spending 15 years at MIT's Lincoln Laboratory, joined BBN in late 1966, led the ARPANET team at BBN in the late 1960s and the 1970s, and retired in 1994 as president of BBN's Systems and Technology Division. Heart is a member of the IEEE and has been a member of the IEEE Boston Section executive committee. He was a member of the IEEE's predecessor IRE, and, representing the IRE, was a founding director and treasurer of the American Federation of Information Processing Societies. He led the BBN engineering team for which BBN received an IEEE Corporate Innovation Recognition award in 1999. Heart served two terms as a member of the USAF Scientific Advisory Board.

Steve Levy is the general partner of Levy Venture Partners. In 1995, he retired as chairman from BBN, which he joined in 1966. From 1976 to 1994, he was BBN's CEO. Levy, who has served on many corporate boards, is past chairman of the American Electronics Association (AEA), the Massachusetts High Technology Council, and the Massachusetts Telecommunications Council. He served on the U.S. Department of Defense's Policy Advisory Committee on Trade and the AEA's National Information Infrastructure Task Force. He holds a BBA in accounting and was awarded an honorary doctor of laws degree from the University of Massachusetts.

John Makhoul joined BBN in 1970. Currently, he is working on various aspects of speech and language processing, optical character recognition, and human-machine interaction using voice. He is also an adjunct professor at Northeastern University and Boston University. Originally from Lebanon, Makhoul received degrees in electrical engineering from the American University of Beirut, Ohio State University, and MIT. Makhoul is a Fellow of the IEEE and of the Acoustical Society of America. His awards include the IEEE's Third Millennium Medal.

Alex McKenzie worked at BBN from 1967 to 1996 in a variety of positions related to network design, implementation, and management. As a BBN manager he was responsible for 250 staff members and an annual budget over $50 million. He helped develop communication protocols in the ARPANET/Internet Network Working Group, the IFIP working group on Computer Networks (chair 1979-1982), and the International Organization for Standardization (chair of Presentation Layer group). He was awarded the IFIP "Silver Core" for outstanding service in 1986. Alex received a BS from Stevens Institute of Technology, an MS from Stanford University, and a certificate from the Sloan School of Management at MIT.

Ray Nickerson joined BBN in 1966 and was with the company, in Cambridge, as a researcher and manager, for 25 years. He was director of the Behavioral Sciences Division (which became the Information Sciences Division in 1975) or deputy director of the Computer and Information

Sciences Division for most of the period covered by Chapter 8. He retired as a senior vice president of BBN Systems and Technologies in 1991 and is now a research professor at Tufts University.

Craig Partridge is Chief Scientist for Internetworking at BBN Technologies, where he has worked on data networking problems since 1983. He is best known for his work on email routing, TCP round-trip time estimation, and high-performance router design. He received an MSc and a PhD, both in computer science, from Harvard University. Partridge is the former editor in chief of IEEE Network Magazine and ACM Computer Communication Review and is an IEEE Fellow.

Rick Schantz is a principal scientist at BBN Technologies where he has been a key contributor to advanced distributed computing research since joining the company in 1973. His research has been instrumental in defining and evolving the concepts underlying middleware since the early days of the ARPANET and Internet. More recently, he has led research in developing and demonstrating the effectiveness of middleware support for adaptively managing real-time, end-to-end quality of service and system survivability. Schantz received a PhD in computer science from the State University of New York at Stony Brook. He is a Fellow of the ACM.

Ed Starr joined BBN in 1959. He initially worked in the physical sciences and moved to computer systems a decade or two later to work on the Butterfly Multiprocessor. He assembled excellent teams and led them to accomplish large, difficult programs such as the Defense Data Network (DDN) for the Defense Communications Agency and the Fixed Distributed System (FDS) for the Navy. He retired as General Manager and CEO of BBN in 2001.

John Swets is a healthcare policy lecturer at Harvard Medical School and a radiology research associate at Brigham and Women's Hospital. He retired in 1998 from BBN after serving as senior vice president; general manager of research, development, and consulting; and board of directors member (all from 1970–1974); and chief scientist for information sciences (1975–1998). Previously, he taught at MIT. He also taught at the University of Michigan, from which he earned a BA and a PhD in psychology. Swets is a member of the National Academy of Sciences and the American Academy of Arts and Sciences. His awards include the Warren Medal of the Society of Experimental Psychologists.

Dave Walden worked at BBN from 1967 to 1970 and from 1971 to 1995, serving in a variety of technical, management, and staff positions. Walden was a member of the engineering team for which BBN received an IEEE Corporate Innovation Recognition award in 1999: "For pioneering contributions to computer networking technology through the development of the first packet switches, the ARPANET Interface Message Processor (IMP) and Terminal Interface Message Processor (TIP)." He has written extensively on technical and management topics and on computing history.

Ralph Weischedel is a principal scientist and heads the natural language processing group in the Speech and Language Processing Department at BBN. He holds a PhD in computer and information sciences from the University of Pennsylvania. Prior to joining BBN in 1984, he was a tenured associate professor at the University of Delaware. Weischedel is a former president of the Association for Computational Linguistics.

Part I
Founders and Early Days in Computing

This first part of this volume describes the early days of BBN and BBN's entry into computing. In Chapter 1 BBN cofounder Leo Beranek describes founding the company and recruiting psychologist J. C. R. Licklider to BBN, for the purpose of moving BBN toward computers. In Chapter 2 Ray Nickerson provides a sketch of the other BBN cofounder, Dick Bolt. In Chapter 3 John Swets describes the involvement of Licklider and other psychologists in BBN's move into computers and in the early world of computing more generally. In Chapter 4 Dave Walden has compiled material on the basic computer systems BBNers built in those early days.

≈ ≈

Two of the authors of chapters in this part of the book have published memoirs:

- Leo Beranek, *Riding the Waves: A Life in Sound, Science, and Industry*, MIT Press, Cambridge, MA, 2008.
- John Swets, *Tulips to Thresholds: Counterpart Careers of the Author and Signal Detection Theory*, Peninsula Publishing, Los Altos Hills, CA, 2010.

Naturally these early and longtime BBN employees describe the company more generally while describing their own careers at BBN.

Chapter 1

Founding a Culture of Engineering Creativity

Leo Beranek

In establishing BBN, the founders deliberately created an environment in which engineering creativity could flourish. The author describes steps taken to assure such an environment and a number of events that moved the company into the fledgling field of computing.

1.1 Introduction

During World War II, I served as director of Harvard University's Electro-Acoustic Laboratory, which collaborated with the nearby Psycho-Acoustic Laboratory (PAL).[1] The daily close cooperation between a group of physicists and a group of psychologists was, arguably, unique in history. One outstanding young scientist at PAL made a particular impression on me: J. C. R. Licklider, who demonstrated an unusual proficiency in both physics and psychology. Another individual, a psychologist, who distinguished himself at PAL was Karl D. Kryter. I made a point of keeping their talents close by in the ensuing decades, as they would ultimately prove vital to the growth of Bolt Beranek and Newman Inc. (BBN) in the upcoming man-machine symbiosis age.

In 1945, at the close of World War II, Richard Henry Bolt became an associate professor of acoustics in the Physics Department at the Massachusetts Institute of Technology (MIT). With Bolt as its director, a new acoustics laboratory was immediately formed, which had faculty supervisors from the fields of physics, electrical engineering, architecture, and mechanical and aeronautical engineering.

Two professors at MIT were then world leaders in acoustics, Philip Morse and Richard D. Fay. They, along with Bolt and MIT President Karl Compton, enticed me away from Harvard in 1947 with the title of associate professor in communication engineering (tenured) and technical director of the Acoustics Laboratory. The laboratory was financed primarily by funds from the U.S. Navy's Bureau of Ships, although there soon was additional financing from the Office of Naval Research. I began teaching a course in September 1947 called, appropriately, Acoustics. My office was across the corridor from Bolt's, and our contracts with MIT allowed each of us one workday a week, plus weekends and summer, to do personal consulting.

1.2 BBN's beginnings

Requests regularly came into the office of MIT's president asking for acoustical help. Those requests were routinely routed to Bolt. One arrived in 1946 from the New York architectural firm, Harrison and Abramowitz, requesting a quotation for services as potential consultant to the United Nations permanent headquarters to be built in New York City. Dick bid and won the commission. In October 1948, a set of drawings for the project arrived, which, when unrolled on his office floor, was 8 inches thick and 10

Timeline

1. Bolt Beranek partnership formed October 15, 1948, and by 1949 had five employees.
2. Moved to 57 Brattle Street in October 1949.
3. Newman admitted to partnership in 1950 and name changed to Bolt Beranek and Newman.
4. Moved to 16 Eliot Street in 1951.
5. Sam Labate and Jordan Baruch admitted to partnership in January 1952.
6. In late fall of 1953, the start of a formal organization began, with Labate as administrative assistant. I moved my office from the Acoustics Laboratory at MIT to Eliot Street, and at MIT moved to smaller space in Building 10.
7. K-Plan was instituted in January 1953.
8. The Blen Corporation, a subsidiary, was formed in 1952.
9. BBN incorporated in December 1953, and BBN transferred its government contracting from "fixed overall price" to "cost plus fixed fee."
10. Los Angeles office opened in 1956 with three employees.
11. In January 1957, Cambridge headquarters moved to 50 Moulton Street (24,000 sq. ft.), with 66 employees.
12. In 1957 BBN added man-machine and information systems, hiring Licklider and Kryter.
13. Hospital-Medical activities started in 1959.
14. Added a second building in 1960 (32,000 sq. ft.). The Cambridge office now employed 148 and the Los Angeles office employed 22. A Chicago office opened with 3 employees in 1960.
15. Prototech, Inc., with Walter Juda a president, was added as subsidiary in mid-1961 with fuel cell development as its principal activity.
16. BBN's Initial Public Offering, was made June 27, 1961. Then, Baruch resigned as treasurer to devote his time to hospital-computer activities and John Stratton, an MBA graduate, replaced him, becoming the sixth member of the Board.
17. In 1962 the gross income for the different activities was: applied physics — 39%; architectural acoustics and noise control — 28%; instrumentation — 10%; psychoacoustics and psychophysiology — 8%; man-machine and information systems — 12%; and bio-medical technology–3%. Government contracts contributed 52% to the company's gross income.
18. In September 1962, Licklider took a leave of absence to go to ARPA in Washington.
19. A New York office was formed in 1963 and by 1964 it had four employees, while Prototech and Blen together had a total of 23 employees. At this time, Blen Corp. had two divisions: educational products which included teaching machines and advanced study courses in five cities on random processes, oceanography, modern optics and systems engineering; and the Data Equipment Co. that manufactured scientific instrumentation.
20. In 1964 Jerome Elkind and John Swets were elected vice presidents of BBN and co-directors of man-machine information systems. John Senders was elected Principal Scientist.
21. Frank Heart joined BBN in December 1966.
22. Proposal for producing the ARPANET was completed in September 1968.
23. The first two stations of the ARPANET, the "IMPs," were shipped, and the first communication occurred on October 3, 1969.

feet long. Dick realized that the project was more than a one-man job, and he called me in to share his awe. Dick immediately proposed that we form a partnership — we had papers drawn up some days later — and Bolt and Beranek came into existence (see Figure 1.1).

Bolt had received his PhD in acoustics from the University of California at Berkeley in June 1939. He was a dynamic, 5 feet 11 inches tall, handsome man with a ready smile

Figure 1.1. Partners Leo Beranek and Dick Bolt, summer 1949. (Photo from author's personal collection.)

and brilliant mind. He had the ability to quickly absorb new fields and become adept at understanding and working in them. At MIT he was a popular lecturer and attracted many promising students into the field of acoustics. He was a judicious, thoughtful administrator and was liked by all who came into contact with him. His relation to me was always excellent, with hardly ever any misunderstanding.

The firm, Bolt and Beranek, had the blessing of MIT's new president, James Killian. He offered to help us get started and rented us two rooms in the MIT Acoustics Laboratory for our use, but warned us that we would have to seek space outside of MIT if our needs expanded. Our first employees, each part time, were four brilliant MIT students working for their graduate degrees: Robert Newman, Jordan Baruch, Samuel Labate, and William Lang. Other consulting requests came to MIT, and we soon had to buy acoustical measuring equipment, which took up all the space in the two rooms.

A little over a year later, Bob Newman completed his architectural degree. In relatively short order, we employed him and in 1950 changed the partnership's name to Bolt Beranek and Newman (BBN). Newman had received his master's degree in physics at the University of Texas and, during World War II, had worked for two years at Harvard's Electro-Acoustic Laboratory and for the remaining part of the war at a naval research laboratory in Pennsylvania. At the end of the war, he enrolled in a graduate school program in architecture at MIT. Bob was short, about 5 feet 5 inches, and had a good eye for architectural design. He quickly learned the basics of architectural acoustics from Bolt and me and soon was in charge of BBN's architectural acoustics division. As a lecturer to architects on acoustics, he was a master. Every architect who attended his lectures at MIT — as well as at Harvard and a dozen other top universities — vividly remembers both him and what he taught.

Returning to the United Nations project: it was very demanding. The architect, Wallace Harrison, produced a design for the General Assembly building that was a large truncated cone. The U.N. delegates sat at tables on the floor of the cone facing the cone's north wall. A large two-level seating space for an audience was attached to the cone, projecting externally, on the south side. Bolt and Newman took responsibility for the acoustical treatment and encountered no unusual problems. The sound system design was left to me, and it proved to be almost unsolvable. Near the slanting north wall of the cone, on a raised platform opposite the audience seating, is a bench for about three people. The Secretary General of the United Nations and his staff generally sit there. Between that bench and the seating for the delegates is a podium from which all formal speeches are made. The sound system had to cover both the delegates on the main floor and the visitors in the audience space. This meant a large and multi-element loudspeaker system.

My immediate recommendation was to hang the loudspeaker array over the podium, perhaps camouflaged by a surrounding, transparent world globe. Harrison would have none of that and stipulated that it must be in the wall behind the bench. This meant that the loudspeakers would be behind and above the shoulders of the person speaking at the podium. This is a sure formula for acoustical feedback. In desperation, I sought out loudspeakers and microphones that had the least phase shift in the frequency range between 300 and 4000 Hz and fortunately I found them at the Altec Lansing Company. A recessed space above the Secretary General's bench was built to contain the loudspeakers. Covered with an acoustically transparent surface, they are invisible. I had the array of loudspeakers mounted to serve the various audience areas and then I had the space around and between them filled with a highly-sound-absorbing acoustical material, which killed any possible acoustical cavity resonances. Those sitting directly in front of the podium were served by a loudspeaker mounted in the podium's front. Miraculously, this limited-frequency system worked without any feedback and the speakers' voices were perfectly intelligible.

In the total U.N. complex, we prescribed the acoustics for the many meeting rooms (e.g., the Security Council room), and they all were successful. This prestigious success made our name known to architects everywhere, and our business boomed.

In 1949, I convinced MIT's Department of Electrical Engineering to appoint Licklider as a tenured associate professor and to work with me in the Acoustics Laboratory on voice communication problems. A new office was built for him on the floor above mine. Shortly after his arrival, he being the only psychologist on the MIT faculty, the department chair asked Licklider to serve on a committee that established the Lincoln Laboratory, an MIT research powerhouse supported by the Department of Defense. The opportunity introduced Licklider to the nascent world of digital computing, although he had no occasion to work with, or to learn programming on, their two new experimental machines, the TX-0 and the TX-2. Licklider devoted a fair amount of his time to Lincoln Lab projects, one example being his help in the lab's discovering that airplane identification by radar signals could be improved through measuring the reflected signal's modulation by the (audio) frequency of the rotating propellers. In addition, in the Department of Economics and Social Sciences of the School of Humanities, Licklider hired a number of promising young psychologists, the first of whom was George Miller in 1951, in an effort to form a psychology department at MIT. It seems that this group grew without the formal knowledge of the Dean of the School of Humanities. As a result, the administration later killed Licklider's plan for a psychology department. Thus, for much of the time, the Acoustics Laboratory only benefited from about one-half of his efforts.

1.3 Steady growth and expansion

BBN's business grew steadily and more staff was rapidly added. In October 1949, we vacated the MIT space and moved to the second floor of a (now nonexistent) building at 57 Brattle Street. In 1951, we moved into two apartments and the basement of a six-apartment building at 16 Eliot Street in Cambridge (see Figure 1.2). We also opened an office in Los Angeles. In the next few years, we took over additional apartments and by 1955 we occupied the entire Eliot Street building. In 1956, we boasted 50 full-time employees plus several part-time employees or consultants. We moved into an existing building at 50 Moulton Street in Cambridge in 1957 and added a two-story building adjacent to it in 1959. Figure 1.3 shows Labate and Newman at the 1959 groundbreaking for our Moulton Street addition; Figure 1.4 is a recent photo of that facility's entrance.

Figure 1.2. The home of BBN in 1953: 16 Eliot Street, Cambridge, Massachusetts. (Photo from author's personal collection.)

Jordan Baruch and Sam Labate had been with us on a part-time basis almost from the day the company was formed and were admitted to partnership in January 1952.

Jordan Baruch was the most brilliant of my students. He had come to MIT to be an electrical engineer and was a straight-A student. He had taken my acoustics course in 1948, the second year that I taught it. Jordan was one of 160 in the class. He was quick to understand what I was teaching and asked so many questions that, after a week or so, I suggested that he yield more time to others.

Jordan wanted to do a joint-department research project, and he was told that this was only possible in the acoustics laboratory. He elected to have a thesis committee from the departments of electrical engineering, physics, and mechanical engineering. Jordan's father was not well off and Jordan needed financial assistance if he were to continue for a doctorate. I arranged for him to receive the Celotex Fellowship one year and the Armstrong Cork Fellowship another year, or two years. He went on to receive his doctorate in 1950. His thesis was on instrumentation for measuring the transmission of sound through panels.

Jordan had what I would call a photographic memory. For example, at BBN he read a five-volume set of military procurement books in just a few days, yet ably referred to

Figure 1.3. Sam Labate and Bob Newman breaking ground in 1959 for an addition to BBN's 50 Moulton Street building in Cambridge, Massachusetts. (Photo courtesy of BBN Technologies.)

Figure 1.4. Entrance to BBN's 50 Moulton Street building, Cambridge, Massachusetts, in 2005. (Photo courtesy of Jennie Connolly.)

almost any part of the text when discussing the contents with government contracting personnel. In addition, he was well informed about a wide variety of subjects, such as health, gardens, automobiles, and computers.

Samuel Labate had come to MIT after World War II to study in the mathematics department. He took my acoustics course and became acquainted with Dick, me, and the staff at the Acoustics Laboratory. Sam's master's thesis was on measurement of acoustic materials using an impedance tube. He proved to be a clear thinker and was well liked by his fellow students and supervisors. Because of Sam's "can do" attitude, he was a valuable and adaptable acoustical consultant who could be depended on to carry a job through to completion.

Bolt, Newman, and I had discussed at some length how we wanted the company to grow. Since Baruch was a highly trained acoustical engineer, learned easily, and seemed our equal in every way, the decision to make him a partner was straightforward. Labate was a less well trained engineer and tended to be more interested in the business aspects of the firm, and we had longer discussions about asking him to join the partnership. We notified both of their partnership on the same day. Baruch was not surprised, but Labate said to me afterwards that he never dreamed that we would include him.

In December 1953, BBN incorporated, the primary reason being to isolate the partners from liabilities that came from an important area of business: the control of jet aircraft noise. Just as we had begun operations, we had been contacted by the National Advisory Committee on Aeronautics and by companies engaged in the manufacture of jet engines. These organizations had asked us to design structures for testing engines that would minimize noise. With BBN's incorporation, Bolt was named chairman of the board, I was president and CEO, Labate was executive vice president, Newman was vice president, and Baruch, treasurer.

1.4 Finances

The five partners owned all the stock in equal amounts and constituted the entire board. This created a concern on our part that the high-level people we were employing would become restless if all the financial profits from their work accrued to only five people. Thus we devised several somewhat novel means to alleviate this worry and reward employees.

First, we instituted the K-factor plan to inflate the salaries of key personnel. The K-factor was formulated by determining the ratio R of the company's total gross income to its total salaries and inserting it in the formula $K = 0.66 + 0.33R$. The basic salary of each participant was multiplied by the K-factor. The value of K was limited to the range 0.75 to 1.5. For many years the K-factor varied only from 1.1 to 1.2.

The second means of reward was to establish a stock purchase plan. The purchase price was set at the beginning of a year by the book value of the company, and the participant had to pay for the stock within 12 months. This led to a handsome gain when the company went public in 1961 (see Figure 1.5).

The third means was to establish a promotion structure for technical personnel that paralleled the conventional corporate ladder — for example, a typical corporate progression was unit head, division head, vice president, president, and CEO. In our parallel technical ladder, the first step was the title consultant, engineer, or scientist (C, E, or S). The next step was senior C, E, or S. Third was principal C, E, or S. In 1969, we established the title of chief C, E, or S. Salaries at the various levels were commensurate with the salaries of the administrative heads. Above all, I insisted that the motto of the company be, "Each new person hired should raise the average level of competence of the firm." This became an operating creed that kept us from hiring anyone who we believed was not as smart as ourselves.

Figure 1.5. BBN IPO day: Leo Beranek, Jordan Baruch, Dick Bolt, Samuel Labate, and Robert Newman, summer 1961. (Photo from author's personal collection.)

The company grew without the help of outside financing, except for maintaining a line of credit at the First National Bank of Boston. By 1961, with borrowings of $325,000, it was apparent that the company needed cash for expansion, and it went public, raising nearly $1 million. Baruch as treasurer and I as CEO planned the offering, working with our auditors and lawyers. An interesting point was our selection of the underwriter for the offering. We interviewed several investment firms: Paine Webber thought our offering price should be $4.50 per share; Smith Barney thought $8.50; but we chose Hemphill Noyes & Co., which took us public at $12 per share. The price on opening day rose to $18. It remained above the level of $12 well beyond the next year.

Two things contributed to the eventual fall from the "balloon" price. First, the financial pages of the newspapers began publishing price earnings (P/E) ratios, and our stock immediately lost some value. Worse was that the *Wall Street Journal* actually named Bolt Beranek and Newman as a company with an overly valued stock and after that its value soon dropped to $4.50 a share.

1.5 New directions: Licklider and computers

As president, more and more of my time was taken up by BBN activities. Consequently, I reduced my teaching load at MIT to 75 percent in 1951 and to 50 percent in 1953. I resigned from my tenured professorship in 1958, thereafter teaching two-week summer courses on noise control for several years. Bolt remained a full-time professor, devoting only his one day a week to BBN.

The company's extensive work in developing acoustical criteria for acceptable noise levels outdoors, and in building spaces, resulted in a decision to develop a stronger and broader activity in psychoacoustics — the science of sound as it affects humans. From our initial interest in how people respond to aircraft noise, we were led to other aspects of psychoacoustics, notably human speech and hearing. Our company obtained government contracts to support research on speech compression, criteria for predicting speech intelligibility in noise, and last, but certainly not least, the reaction of communities around airports to propeller-aircraft noise.

At first, this research was done largely by acoustical engineers who made neighborhood surveys and conducted some experiments with the aid of analog computers that, for example, presented subjects with pairs of signals between which they had to choose. Two professors from MIT were employed part-time as consultants to give this effort a solid research basis: Kenneth Stevens, a speech expert, and Walter Rosenblith, a physiologist.

Around 1955, I began seriously to consider the long-range directions of the company. My thoughts were guided by my experience in World War II with the psychoacoustic personnel at the Electro-Acoustic and Psycho-Acoustic Laboratories at Harvard and, later in the war, by my experience as head of the Systems Research Laboratory in Jamestown, Rhode Island. The mission of that facility was to speed up the handling of information on U.S. warships so that they could more effectively combat the mounting danger of Japanese kamikaze aircraft.

I visualized a potential growth region for BBN in man-machine systems, machines that efficiently amplify human labor. Two examples were in my mind: optimization of aircraft blind-landing, and the performance of racing boats (for example, in the America's Cup race). I reviewed my knowledge of people working in areas at BBN related to this, and Licklider loomed as the outstanding candidate. He not only was a first-rate psychologist with physics training, but at MIT he had acquired considerable knowledge about the uses of computers, through his exposure to the Semi-Automatic Ground Environment (SAGE) air defense system and from Lincoln Lab's computer gurus Wesley Clark, Jay Forrester, Kenneth Olsen, and Ben Gurley.

A look at my appointment book from those days shows that I courted Licklider with numerous lunches in spring 1956 and one critical meeting in Los Angeles that summer. Because joining BBN meant that Licklider would have to give up a tenured faculty position at a major institution, we persuaded him by offering a rather large stock option at about $1.50 a share and the title of vice president in charge of man-machine and information systems. Licklider came aboard in the spring of 1957.[2,3,4]

Lick, as he insisted that we call him, was outgoing and always on the verge of a smile (see Figure 1.6); he ended almost every second sentence with a slight chuckle, as though he had just made a humorous statement. He walked with a gentle step, often with a Coca-Cola in hand, and he always found the time to listen to new ideas. Relaxed and self-deprecating, Lick merged easily with the talent already at BBN. He and I worked together especially well: I cannot remember a time when we disagreed.

Licklider had been on staff only few months when he told me, in fall 1957, that he wanted BBN to buy a digital computer for his group. When I pointed out that we already had a punched-card computer in the financial department and several analog computers in the experimental psychology group, he replied that they did not interest him. He wanted a then state-of-the-art digital machine produced by the Royal-McBee Co., a subsidiary of Royal Typewriter.

"What will it cost?" I asked.

"Around $30,000," he replied, rather blandly, and noted that this price tag was a discount he had already negotiated.

I exclaimed, "BBN has never spent anything approaching that amount on a single research apparatus. What are you going to do with it?"

"I don't know," Lick responded, "but if BBN is going to be an important company in the future, it must be in computers."

Although I hesitated at first — $30,000 for a computer with no apparent use seemed just too reckless — I had a great deal of faith in Lick's convictions and finally agreed

Figure 1.6. Louise and J. C. R. Licklider, December 1954. (Photo from author's personal collection.)

that BBN should risk the funds. I presented his request to Labate and Baruch, and with their approval, Lick brought BBN into the digital era. Lick sat at that computer many hours each day, literally hoarding the machine, learning how to do digital programming.

Licklider hired Karl Kryter (see Figure 1.7) in October 1957, and he became actively involved in speech bandwidth compression and effects of noise on sleep. Soon after, in 1958, Thomas Marill (interested in auditory signal detection and artificial intelligence) and Jerome Elkind (interested in the man-machine area) joined BBN.

1.6 Men and machines

Our 1958 client brochure stated that BBN's Engineering Psychology Department had two divisions: one for communication studies that served to identify man's capabilities in the establishment and control of information flow, whether between humans or between men and machines, and another for man-machine studies that served to establish the engineering criteria for the optimum design of a man-machine system, whether a factory, vehicle, or computer.

Within a year of the computer's arrival, in fall 1958, Ken Olsen, president of the fledgling Digital Equipment Corporation, stopped by BBN, ostensibly just to see our new computer. After chatting with us and satisfying himself that Lick really understood

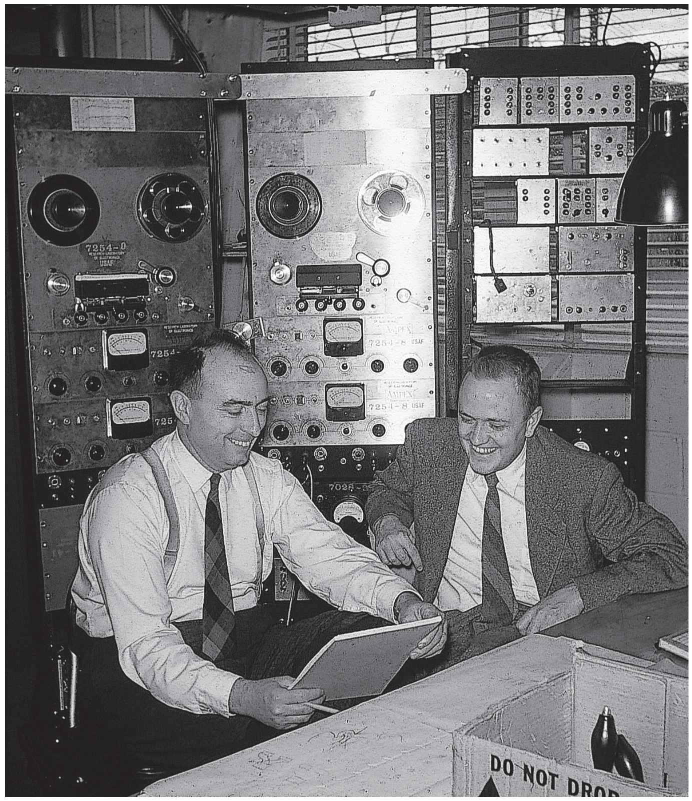


Figure 1.7. Karl Kryter with David Green in laboratory at BBN's 50 Moulton Street, Cambridge, Mass., building, 1958. (Photo from author's personal collection.)

digital computation, he asked if we would consider a project. He explained that DEC had just completed construction of a prototype of its first computer, the PDP-1, and that they needed a test site for a month. We agreed to be a test site, at our regular hourly rates.

The prototype PDP-1 was a monster compared to the Royal-McBee; it would fit no place in our offices except the visitors' lobby, where we surrounded it with Japanese screens. Lick and Ed Fredkin — a youthful and eccentric genius who came to BBN because of the Royal-McBee in 1958 — and several others put it through its paces for most of the month, after which Lick provided Olsen with a list of suggested improvements, especially how to make it more user friendly.

The computer won us all over, so BBN arranged for DEC to provide us, in 1960, with its first production PDP-1 on a standard lease basis. Then Lick and I took off for Washington, D.C., to seek research contracts that would make use of this machine, which carried a price tag of $150,000. Our visits to the Department of Education, National Institutes of Health, National Science Foundation, NASA, and the Department of Defense proved Lick's convictions correct, and we soon secured several important contracts.

In 1961, Lick hired William Neff as head of a biomedical unit, assisted by Philip Nieder and Norman Strominger. Their first government contract was for research on basic brain function and behavior, in particular, neuromechanisms of hearing. At the same time, he hired Vincent Sharkey to work on human factors. Sharkey's contract support was mostly highly classified.

By winter 1961, our client brochure divided BBN's human related systems activities into five parts:

- man-computer symbiosis (time sharing, light-pen control by touching the monitor, and real-time control of research studies)
- artificial intelligence
- man-machine systems (information displays, audible signals to supplement visual radar information, and pattern recognition)
- psychoacoustics and psychophysics (intelligibility and naturalness of speech in communication systems, speech compression, deafening effects of impulse noise, brain wave responses of man to sound, and noise during wakefulness and various stages of sleep)
- biomedical research (colorimeter of digital type, and instrumentation for recording and displaying the physiological variables and visual signals needed by surgeons during open-heart and brain surgery)

Also, we listed engineering psychology under the direction of John Senders, who joined BBN in 1962. He became involved in experiments on the effects of distractions on performance suffered by airplane pilots and automobile drivers.

Once we had the PDP-1, in 1960 Lick brought two MIT consultants on computers into BBN's life, John McCarthy and Marvin Minsky. McCarthy had conceived of time-sharing computers and had pled with MIT computer people to implement the concept, which they were slow to do. At BBN, he found a response in Lick and, in particular, in Ed Fredkin. Fredkin insisted that "timesharing could be done on a small computer, namely, a PDP-1." McCarthy recalled in 1989:[5]

> I kept arguing with him. I said "Well, you'd have to . . . get an interrupt system." And he said, "We can do that. You'd have to get some kind of swapper." I said "We can do that."

An interrupt system enables an external event to interrupt computations that are in progress, and a swapper has to do with swapping among computational streams.

The team, largely led by Sheldon Boilen, created a modified PDP-1 computer divided into four parts, each assigned to a separate user. In fall 1962, BBN conducted a public demonstration of time-sharing, with one operator in Washington, D.C., and two in Cambridge. To augment the PDP-1's small memory, BBN acquired the first FastRand rotating drum, made by Univac, with a 45-Mbyte storage capacity and an access time of about 0.1 second. (For more about this early time-sharing system, see Chapter 4.)

Under Jordan Baruch's direction, BBN installed a time-shared information system in winter 1962 in the Massachusetts General Hospital that allowed several nurses and doctors to create and access patient records at a number of nurses' stations, all connected to our central computer (see Chapters 4 and 12).

1.7 New directions in psychology

In 1961 and 1962, Licklider was heavily involved in the "libraries of the future" project. (Full details of this project are presented by John Swets in Chapter 3.)

In summer 1962, Lick was lured by Jack Ruina, director of the Advanced Research Projects Agency (ARPA), to go to Washington in October to head up its Information Processing Techniques Office.[6] Swets joined BBN in 1962 to take over the library project, and Senders also joined the effort. Licklider wrote the final report from Washington, in the form of a book, *Libraries of the Future*,[7] with chapter assistance by Daniel (Danny) Bobrow, M. C. Grignetti, John Swets, Tom Marill, and John Senders. This report was distributed to libraries widely and has been influential in pioneering the use of computers in libraries.

In the early 1960s, new activities in engineering psychology were pursued. For example, BBN was awarded a NASA/U.S. Air Force contract to determine the capacity of pilots to perform and adapt under flight conditions that change quickly and in complicated ways; to recommend display requirements for information essential in the Apollo Manned Space Vehicle System; and to the use of computers in education. In the artificial intelligence area, BBN's ongoing work involved recognition of patterns, memory organization, and machine language. Additionally, Swets carried out studies on the Socratic teaching method, and Baruch continued work on the Massachusetts General Hospital time-shared system.

In 1966, BBN had two software projects that vitally needed outside help: the hospital project and a computer system planned for the company-wide use of a large firm in the Boston area. Bolt and Bobrow convinced Frank Heart that he should come aboard to head up the information sciences and computer systems division of BBN. Ray Nickerson also joined that year, working with Jerry Elkind.

1.8 ARPANET

Then came ARPA's request for proposals to build the ARPANET in August 1968. Heart was selected to manage the response and he put together the Interface Message Processor (IMP) group. The proposal was submitted in September 1968. ARPA responded with a $1 million contract, and the first IMP was completed and shipped to the University of California, Los Angeles, in September 1969. Others followed monthly. The second IMP was shipped to the Stanford Research Institute, and on 3 October, the first message on the two ARPANET stations was sent: LO — phonetically, "ello."

The work on the ARPANET coincided with the return of Dick Bolt to the company. For over a decade, BBN had been deprived of his services. He had left the company and MIT in 1957, following the nonrenewal of a government research contract at the MIT Acoustics Laboratory. After his departure, he was appointed by the National Institutes of Health to be the principal consultant in biophysics to work with a new study section in that field. Three years later, he was named associate director of the National Science Foundation, also for a three-year stint. The following year he was a Fellow of the Center for Advanced Study in Behavioral Sciences at Stanford University. On his return to MIT, he served for several years as a lecturer in the Department of Political Science. He served BBN until he retired in 1976, and resigned from the board of directors in 1981.

1.9 Thoughts on managing BBN

A novel management feature, applicable to a research organization, but not a manufacturing company, was inaugurated by me in about 1957. It had been my observation that a lot of time can be spent by a researcher or a consultant on problems related to money. Also, it was becoming essential to have tighter controls on chargeable time, billing of clients, and better communication with the financial office. To satisfy these growing demands, I set up a financial arm parallel to the research organization.

Under this scheme, each technical department had assigned to it a financial person from this new arm. This person, whom I called a facilitator, had two bosses, the head of the department and the chief financial officer. If a person in a department wanted to buy a piece of new equipment or set up a new research facility, he would sit down with the facilitator and outline his needs. The facilitator would work out with him the specifications on the apparatus and the space needs. Then, after obtaining approvals from the management, the facilitator would attend to the purchasing of the equipment and the location and modification of the desired space. If appropriate, the facilitator would solicit competitive bids. In addition, he made sure that each employee in his department submitted a weekly time sheet, and he kept track of sick and vacation times. He also followed the progress of each work in comparison with its contract and checked against deadlines and penalties. He drafted bills to clients based on the time sheets and the terms of the contract.

The facilitator was required to consult with the chief financial officer and would make sure that the department was following the financial rules of the company and the government. Obviously, he was working both for the department head and the financial officer, which meant that his salary was reviewed by both. In my opinion, this arrangement allowed the technical person more freedom to tend to his activities and not be bothered by red tape. From the financial side, contract provisions and deadlines were being met and billings went out correct and on time. Also, savings arose from competitive bidding. This financial arm remained in place until BBN moved into manufacturing.

My own management style needs analysis. At the start, I was senior in age and experience to all employees, except for Bolt. Through my research and the research of graduate students at MIT, I was a source of new knowledge. This meant that I took leadership in a number of key projects and acted as a close partner with the consulting staff. During this period, Bolt and Newman tended to the architectural acoustics projects that kept pouring in. Labate was responsible for the day-to-day management, and I talked with him every day.

Overall, my management style was to work with the staff whenever possible, to treat the staff as equals, and to make them aware that BBN was a highly professional organization. Licklider exemplified this same style. I held weekly meetings with senior members of the staff to learn what needed to be done to improve our operations. In writing, I encouraged our staff to become members in appropriate technical societies and to write papers for publication. BBN authorized attendance at any technical meeting where an employee was to present a paper, provided the division head said the paper was first class. If no paper was being presented, attendance at one meeting a year was automatic. Attendance at an additional meeting was approved if there was to be a specially informative symposium. This attitude then carried over into the computer work that followed, although I never took part in the technical side of the man-machine and psychoacoustics endeavors.

≈ ≈

After my participation in developing the sound system for the General Assembly Hall in the United Nation Headquarters, I took on major responsibility for quieting the supersonic wind tunnel at the NASA Lewis Laboratory in Cleveland. The purpose of this tunnel was to test special jet engines in supersonic windstreams. When first operated it created so much noise in the surrounding neighborhoods that the City of Cleveland forbade further operation. I was called in on an emergency basis to quiet it. It was a major project and involved techniques that had never been used before — even my partners feared that my designs might not succeed. The result was the largest muffler ever built, 220 feet long, 33 feet wide and 46 feet high. It was completed in 1950 and was a complete success.

The Convair Aircraft Company in San Diego then asked BBN to take responsibility for reducing the noise in the passenger compartment of their new Model 440 propeller-driven passenger airplane. I choose Ed Kerwin as my partner and the two of us with our wives lived in San Diego for two months in the summer of 1954. There, I designed novel mufflers for the exhausts of the two engines and Ed rearranged the exhaust tubes in the engines to further reduce the noise. We also designed a new acoustical lining for the interior and asked for thicker window panes. This job was also a complete success.

The largest consulting job that BBN undertook during the first decade of its existence was for the Port of New York Authority. The PNYA wanted all new jet-propelled passenger aircraft to create no more noise annoyance in neighborhoods surrounding Idlewild (Now JFK) airport than was created by existing propeller-propelled passenger aircraft. The management of PNYA asked me to take responsibility for this project based on my successes at Cleveland and San Diego. I asked Karl Kryter to take over the responsibility for determining how much the noise from jet-engines had to be reduced to make them no noisier (to listeners) than propeller-engines. Laymon Miller was put in charge of making noise measurements of propeller aircraft in neighborhoods around Idlewild both daytimes and nighttimes. The first jet passenger aircraft was being built by Boeing Aircraft and was to be purchased by Pan American Airways. Measurements were performed by the staff of BBN of the noise produced by this first Boeing 707 aircraft while it was taking off and flying over neighborhoods. Kryter exposed human subjects to the measured 707 noise and the measured propeller aircraft noise and it was found that for equal “perceived noisiness,” the 707 noise would have to be reduced by 15 decibels — a tremendous amount. Boeing was forced to put mufflers on the aircraft. In addition, to bring the 707 noise in neighborhoods down to the desired “equal” level, the plane on takeoff had to climb as steeply as possible; and at about 1.5 miles from start of take-off roll the engine power had to be cut back, and the plane had to fly at a constant altitude until it ceased to be over thickly settled neighborhoods. Boeing, Pan American and even the FAA tried every way possible to get these requirements nullified, even threatening to sue BBN for “incompetence.” But PNYA stood its ground, and the noise requirements went into effect. The first jet passenger aircraft flying out of Idlewild began operations in November 1958, with no objections from surrounding neighborhoods.

I also took on management responsibility for the acoustics of a series of concert halls — usually working with a staff member from the acoustics department. I traveled to hear music in about fifty halls, and I interviewed about two dozen leading conductors and musicians and a wide range of music critics in the USA and England. The halls I was involved in included the Aula Magna in Caracas Venezuela (1954), the Fredric Mann Auditorium in Tel Aviv (1957), the Binyanei Ha’Oomah hall in Jerusalem (1960), the Tanglewood Music Shed (1959), and the Lincoln Center concert hall and opera house in New York (1962–63). This sequence led to my book, *Music, Acoustics, & Architecture*.[8]

≈ ≈

By 1962, BBN had grown to such a size that all my attention was consumed by management activities. After BBN went public in 1961, John Stratton, the new treasurer, began exerting a new influence that almost had grave consequences for BBN. First, he had the idea that BBN should grow by acquisition, rather than at the 26 percent compound annual growth that had occurred up to then (and continued through my presidency, which ended in 1969). Several small companies were acquired by BBN, mostly by an exchange of stock, but all failed.

Then Stratton had his big idea in 1968. He became acquainted with the Graphic Controls Corporation in Buffalo, New York, which made Codex charts and business forms and offered computer services. Its gross income and profits before taxes were about equal in magnitude to those of BBN. He worked out a merger agreement in which BBN would be the surviving company but with a new name, with head-quarters in Cambridge or Boston. The chairman and CEO would be the then-president of Graphic, and the president of the new corporation would be Sam Labate. Stratton would be the executive vice president and chief financial officer. I would become the chief scientist. I particularly remember Jerry Elkind coming to me and expressing his concern about the danger of losing many of our superior personnel if the merger took place. For a variety of reasons, including my objections to the idea, the merger was terminated officially on 26 February 1970.

I was happy to become aware of Frank Heart's capabilities, and I learned more about his interests and activities than almost anyone else in the computer group. From the time of his arrival in December 1966 until the request for proposal on the ARPA network in 1968, he built up the group of researchers that won the ARPA contract, developed the ARPA network, and initiated the age of the Internet. Frank was the only software expert I ever met who could estimate the length of time it would take to complete a proposed project and fall within the expenditures that he had "guesstimated" at the start.

My tenure as president ended in the fall of 1969 and I remained for two years as chief scientist. Labate became president and CEO, Swets was named general manager of BBN, and Nickerson assumed his position as director of the Information Sciences Division. My leaving the office of president was the result of an unexpected development. In December 1962, I had joined a group of 30 men and women who were interested in obtaining a license for the operation of Channel 5, in Boston, a large network-affiliated television station. In 1963, on the application to the Federal Communications Commission, I had agreed to be the president of Boston Broadcasters Inc. with the expectation that the executive vice president, Nathan David, would take over the title if BBI were to get the license. Later, David was involved in a questionable case of stock dealing and he had to resign. So, I was stuck with a new career, and, following extensive newspaper publicity about the station, which identified me as BBI's president, I was pressured by BBN's board to resign BBN's presidency immediately. It was two years before the favorable, final U.S. Supreme Court ruling was received. In the interim, until 1971, I served as BBN's chief scientist.

After a year of hiring and construction, BBI went on the air in March 1972 as WCVB-TV (Channel 5) Boston, with ABC as its affiliated network. Actually, this was a good development for BBN. I could not have managed the digital network business as well as presidents Stephen Levy and George Conrades, and the stockholders did much better under them. In conclusion, WCVB-TV was also a great success, and the New York Times in a lengthy 15 February 1981 article carried the headline, "Some Say This Is America's Best TV Station." It achieved that status through the application of my long-stated premise that "Each new person hired should raise the average level of competence of the organization."

References and notes

1. There are two oral histories of the author at the IEEE History Center, Rutgers University, New Brunswick, NJ: number 320 (conducted in 1996 by Janet Abbate — `http://www.ieee.org/web/aboutus/history_center/oral_history/abstracts/beranekab.html`);
number 454 (conducted in 2005 by Michael Geselowitz — `http://www.ieee.org/web/aboutus/history_center/oral_history/abstracts/beranek454ab.html`).

In 2008 the author published his full length memoir: Leo Beranek, *Riding the Waves: A Life in Sound, Science, and Industry*, MIT Press, Cambridge, MA, 2008.

2. See also Leo Beranek, “Roots of the Internet: A Personal History,” *Massachusetts Historical Review*, volume 2, Massachusetts Historical Society, 2000, pp. 55–75; `http://www.historycooperative.org/journals/mhr/2/beranek.html`

3. For an extensive history of Licklider including his relationship with BBN, see M. Mitchell Waldrop, *The Dream Machine: J. C. R. Licklider and the Revolution That Made Computing Personal*, Viking Press, 2001.

4. The focus of this volume is on BBN’s role in the world of computers, and this article describes my role in moving the company into computers from acoustics. Nonetheless, my own engineering and science work has primarily remained in various areas of acoustics, a field in which I continue to do work today, in my ninth decade: *Acoustics*, Acoustical Society of America, 1986; *Noise and Vibration Control*, Institute of Noise Control Engineering, Iowa State University, 1988; *Concert Halls and Opera Houses: Music, Acoustics, and Architecture*, Springer-Verlag, 2004.

5. “An Interview with John McCarthy,” 2 March 1989, oral history conducted by William Aspray, transcript OH 156, Charles Babbage Institute, University of Minnesota, p. 5.

6. Arthur L. Norberg and Judy E. O’Neill, *Transforming Computer Technology: Information Processing for the Pentagon, 1962–1986*, John Hopkins University Press, 1996.

7. J. C. R. Licklider, *Libraries of the Future*, MIT Press, Cambridge, MA, 1965.

8. Leo L. Beranek, *Music, Acoustics, & Architecture*, John Wiley & Sons, New York, 1962.

Chapter 2

A Sketch of the Life of Richard Henry Bolt (1911–2002)

Compiled by Raymond Nickerson

This chapter presents a brief outline of the life of BBN co-founder Richard Bolt. More details regarding his life and work can be found at www.gbcasa.org/notices/boltobit.html *in an obituary written on the occasion of his death in 2001 by Leo Beranek. This abbreviated account draws heavily from that one, as well as from reminiscences by John Swets, Frank Heart and David Walden.*

Richard Henry Bolt was born in 1911, the son of medical missionaries to China. He married, coincidentally, the daughter of missionaries to China, Katherine Mary Smith, whom he met when both were students at the University of California at Berkeley. He developed an interest in physics after receiving a BA in architecture from Berkeley in 1933 and later decided to pursue graduate training in acoustics, which he did, receiving a PhD from the University of California, Berkeley, in 1939.

In the early days of World War II, after having spent some time as a post-doctoral researcher at MIT and as a faculty member at the University of Illinois, Dick Bolt served as the director of the Underwater Sound Laboratory at MIT. In 1943, he was named Scientific Liaison Officer in Subsurface Warfare to the Office of Scientific Research and Development in London. In 1945, he was appointed Director of a newly established Acoustics Laboratory at MIT. While at MIT he, in collaboration with Leo Beranek, whom he had recruited from Harvard, built what was then the largest acoustics laboratory in the country. The story of the formation of Bolt Beranek and Newman[1] is told in Chapter 1.

Bolt served as the Chairman of the Board of BBN from 1953 until 1957 and again from 1966 until 1976. His resignation as chairman in 1957 was to accept an appointment as Principal Consultant in Biophysics to the National Institutes of Health. This and his subsequent appointment in 1960 as Associate Director of the National Science Foundation kept him at a distance from BBN activities for several years. Later his organizational skills were tapped by a number of agencies and committees to assist in organizing or running meetings and in overseeing the publication of proceedings. A notable case in point was his chairing of the committee of experts that investigated the infamous 18-minute gap on the tape made in President Nixon's office three days after the Watergate break-in.

Upon returning to BBN in the mid 1960s, Bolt once again assumed the board chairmanship and became involved in BBN projects. John Swets remembers him as "a high-level trouble shooter" who would dig into BBN departments or projects whenever he could help with a problem. For example, concern about the Library Project (described in Chapters 1 and 3) led Bolt to design and write a pamphlet for reporting its results.

[1] The third BBN partner, Robert Newman, an architect, had little to do with the company's involvement in computer technology. His role in the company's establishment and development was focused on its activities in architectural acoustics.

He managed the preparation of a proposal to NIH, and oversaw the project, for BBN to conduct a review of the Division of Research Resources, and, with Swets, put together an illustrious panel for a year-long project that included interviews with Institute directors. When Jordan Baruch left the company to establish Medinet (mentioned in Chapters 6 and 12), Bolt became acting director of the Computer Systems Division and subsequently helped recruit Frank Heart for the division director position.

Bolt was a fellow of the Acoustical Society of America, the American Physical Society, the American Institute of Physics and the American Academy of Arts and Sciences, and a founding member of the Institute of Noise Control Engineering. He served as president of the Acoustical Society of America and as the first president of the International Commission on Acoustics. His contributions to the science of acoustics were recognized by the first R. Bruce Lindsay Award and the Gold Medal Award, both from the Acoustical Society of America. His Gold Medal citation read: "For outstanding contributions to acoustics through research, teaching, and professional leadership, and for distinguished administrative and advisory service to science, engineering, and government."

Those of us who knew Richard Bolt primarily because of our connection with BBN — we all knew him as Dick — remember him with great respect and fondness. He was many things — scientist, musician, administrator — but, perhaps more importantly to us, he was a genuinely warm, unpretentious, lively, sensitive and sociable human being. His character was evidenced in many ways, not least of which was the considerable lengths to which he went to care for his beloved wife at home in the face of deteriorating mental abilities in the final years of her life. Until the BBN staff reached 100 or so, the Bolts had the whole company at their house annually for an evening of games and sociability, billed as a Monte Carlo night.

Dick often ate in the company cafeteria. Invariably he would look for a table at which a few people were sitting, join it and liven the conversation. He was equally eager to learn what people were doing and to tell of his most recent projects and ruminations. His interests and energy seemed to be boundless. There was never a sense of being on one's guard because of the presence of a founder of the company, but rather a feeling of interacting with an exceptional, and exceptionally likable, person. Some of us heard him describe his life-long ambition as that of becoming "a jack of all trades and master of one." He was indeed a man of numerous talents and accomplishments; and his mastery extended considerably beyond his chosen specialty of acoustics.

Chapter 3

The ABC's of BBN

From Acoustics to Behavioral Sciences to Computers

John Swets

The discipline of psychology, and specifically the concept of man-machine integration, served to organize computer research and development at BBN, beginning in the 1950s.

3.1 Scope of discussion

This chapter gives a unifying perspective on the history of computer research and development at Bolt Beranek and Newman Inc. (BBN). I suggest that the firm's original focus on A (acoustics) led to its work in B (behavioral sciences, principally psychology) which in turn led to C (its computer activities) — the three areas then existing together. In particular, I suggest that psychological concepts have shaped the company's work on computers from the beginning. In doing so, I treat the first five years of psychology and computers at BBN, beginning in 1958, both by narrative history and project descriptions. Raymond Nickerson and Sanford Fidell chronicle in this volume how the approach to computers from psychology has been evident at BBN since then (Chapter 8). I write as a psychologist on the faculty at the Massachusetts Institute of Technology (MIT) from 1956 to 1962 and on the staff at BBN one day a week from 1958 to 1962, then full-time until 1998.[1]

3.2 Acoustics and psychology at Harvard

Several historians — notably Paul N. Edwards, Katie Hafner, Matthew Lyon, Thomas P. Hughes, and M. Mitchell Waldrop — have noted the influence of psychology on computers and assigned a prominent role to particular developments and people in Cambridge, Massachusetts, during and shortly after World War II.[2] They trace a path from Harvard University to MIT to BBN. I cover the same trajectory with an emphasis on BBN and an inside view there.

At Harvard, psychology and acoustics interacted in the interest of solving military problems of command, control, and communications — in the Psycho-Acoustics Laboratory (PAL) headed by Stanley Smith Stevens and the Electro-Acoustics Laboratory directed by Leo Beranek, later a co-founder of BBN. These laboratories investigated the intelligibility of speech in noisy aircraft, tanks, and submarines as it affected speed and effectiveness of communications. Their conceptual model was the generalized communication system described by Claude Shannon, in which an information source generates a message for a transmitter, which is there converted to a signal for a noisy channel. The signal then is picked up by a receiver that converts it to a message for the destination.[3] All of the elements were covered: human speaker characteristics and training; phonetic composition of oral codes (language engineering); microphone, amplifier, and radio characteristics; and characteristics and training of listeners.[4] The overarching themes were information processing and "man-machine integration."

Prominent among the psychologists at the PAL were research fellows J. C. R. Licklider and George A. Miller. They were able to draw major academic contributions from their applied research on communications, including three chapters in Stevens' era-defining Handbook of Experimental Psychology: Licklider on hearing, Miller on speech and language, and the two in collaboration, on the perception of speech.[5] Licklider went on to lead the application of experimental and cognitive psychology to computers while Miller came to personify the application of communications and computer concepts to cognitive theory in psychology.[6] Licklider's career makes him the lead figure throughout this chapter; I briefly characterize Miller's as well to point up that the communications-centered connection between psychology and computers has been a two-way street — one enjoying the richness of a mutual interaction.

3.3 Harvard's psychology joins MIT's computers

After the war, Beranek moved to MIT as professor of electrical engineering and joined physics professor Richard Bolt, later another BBN co-founder, in the Acoustics Laboratory. Beranek was instrumental in bringing Licklider to MIT in 1949 and Miller followed in 1951. The psychologists variously held appointments in the Acoustics Lab, Electrical Engineering Department, and MIT's off-site Lincoln Laboratory. They were members in good standing of the legendary Research Laboratory of Electronics, the facilitator of multidisciplinary research, and core members of the Psychology section, which was housed in the Department of Economics and Social Sciences of the School of Humanities. Licklider was appointed head of the Psychology section in 1952.

On the main campus, the two men were active in the swirl about Norbert Wiener's cybernetics: the modeling of computational processes in command and control in both humans and machines. At Lincoln, they became acquainted with its new computers: Whirlwind, the first interactive computer, and its heirs, the computers of the Semi-Automated Ground Environment (SAGE) system for air defense, with their multiple display terminals, and the TX-2, the first approximation to a personal computer. And they became acquainted with Lincoln's computer visionaries, including Jay Forrester, Kenneth Olsen, and Wesley Clark.

As leaders of Lincoln's psychology group, Licklider and Miller made contributions to the SAGE system's information displays. In his research, Licklider pursued mainly neurophysiological theories of hearing and the role of humans and machines in complex systems.[7] Miller developed his interests in language, memory, and perception and popularized in psychology Shannon's information theory (used to measure human capacities in terms of bits of information) and the linguistic theory of Noam Chomsky.[8] Both men began thinking about computer models for human cognitive processes and human-computer interaction. The information-processing view of cognitive psychology was coming into view, regarding "humans and animals as cybernetic machines and digital computers."[9]

Information-processing models of the mind and the quantitative use of information theory in psychology were given momentum through three conferences at MIT. Two, on speech communication, in 1949 and 1950, were organized by Licklider and foreign languages head William Locke.[10] The third, the 1956 Symposium on Information Theory, was said to contain all of the core ideas of cognitive science. Talks were given by (among others) Shannon and MIT/Lincoln researchers on information theory, Miller on the information capacity of human memory, Chomsky on transformational grammars, Allan Newell and Herbert Simon on computers' discovering proofs of logic theorems, and Ted Birdsall and me on a decision-making theory of human signal detection.[11] Subsequently, Miller and two of his former PAL colleagues proposed a computer model for human purposive behavior.[12-14]

3.4 From "A" to "B" at BBN

Bolt and Beranek (along with MIT graduate student Robert Newman) cofounded BBN in 1948 to supply consulting services in architectural acoustics. Although Bolt maintained the Acoustics Laboratory at MIT until 1956, Beranek gravitated toward BBN and the company began to develop consulting, research, and development across the full spectrum of acoustics. By 1956 its areas of expertise included auditorium and room acoustics, industrial and aircraft and community noise, speech communication systems, signal processing, noise and vibration in space, and underwater sound. By then, it was quite clearly preeminent as an acoustics organization for its scope and capabilities.

BBN's broad forays into physical acoustics confirmed what its principals already knew: Like a tree falling in the forest needs a listener to make a sound, acoustics plays out through people, as studied in psychoacoustics; it is fundamentally a human-oriented discipline. The intrusion of sonic booms or other jet noise under the flight path, speech from an adjoining office, and heel clicks on the floor above are best measured in "perceived," rather than physical, decibels — thought of as measures of "annoyance." Concert-hall design is replete with subjective effects. Industrial machines, jet aircraft, and space capsules must be quieted, and sonic booms must be largely avoided, for human well being. The speech waveform can be degraded in several ways and remain intelligible to the human. Accuracy of message reception depends to a large extent on the listener's expectations. Sonar operators are taught to derive informative characteristics of targets from their sounds as well as from the sounds' visual representations in spectrograms. And so on.

BBN's beginnings in psychoacoustics, in the mid 1950s, were undertaken largely by two part-time employees from MIT — electrical engineer Kenneth Stevens and physiologist Walter Rosenblith — working with BBN partners Dick Bolt and Jordan Baruch. The projects were directed primarily at community noise around airports.[15]

The BBN principals desired a larger range of psychoacoustics and a contribution from psychologists (behavioral scientists) and Beranek naturally thought of Licklider. Beranek also had in mind a second role for Licklider at BBN, namely, to establish an activity in man-machine integration, as a central thrust in the area of human factors or engineering psychology. Licklider accepted an offer in the summer of 1956 to join the company in the spring of 1957. So the step from "A" to "B" at BBN was taken at that time; psychology was prominently added to acoustics, foreshadowing the contribution of psychology to computers. Licklider's stature in these fields was reflected in his being elected president of the Acoustical Society of America in 1958 and the Society of Engineering Psychologists in 1961.

He was ready to move from MIT, in part, because he felt that the Psychology section he had tried to build was not getting the support of the administration.[16] However, he was apparently becoming entranced by computers at this point and felt he could pursue these interests best at BBN, where he convinced the management to buy a computer (Royal McBee's Librascope LGP-30) the next year.[17]

3.5 Psychology at BBN

To advance the psychoacoustics and human-factors efforts, Licklider recruited (best friend/best man) Karl Kryter, W. Dewey Neff, and Vincent Sharkey — all formerly fellow graduate students of his in psychology at the University of Rochester. Kryter came from a laboratory at Andrews Air Force Base in Maryland (he was earlier at Harvard's PAL) to develop programs in speech and effects of noise.[18] Neff came from Indiana University to pursue his work in "physiological acoustics," with cats and monkeys as experimental subjects.[19] Sharkey came from the Air Force Cambridge Research Center to develop work on human factors.[20] Ken Stevens began contributing to the speech effort then and continued to do so for several decades.

To begin work specifically on man-machine integration, Licklider was joined by two of his former doctoral students at MIT. Thomas Marill, with a doctoral thesis on auditory signal detection, had since worked for two years evaluating the SAGE system at Lincoln Laboratory. Jerome Elkind, who held an electrical engineering undergraduate degree from MIT and received an interdepartmental Sc.D. with a thesis on human tracking behavior (manual control), had spent the previous few years at an RCA laboratory in the Cambridge area. (We can note that both theses were outgrowths of the concern for the human factor in warfare.) Marill did fundamental work in computers at BBN, particularly in artificial intelligence (described below), and managed its first computer department. Elkind created a research activity in human control processes that continues today (see Baron in this volume, Chapter 9) and from 1964 to 1969 he largely managed — though he and I were nominally co-directors — a division that included the by-then-several departments in computers and psychology.

David Green and I joined BBN one day a week in 1958 while assistant professors of psychology at MIT. Licklider had brought us to MIT from the University of Michigan, where we conducted doctoral theses in visual and auditory signal detection, in the psychology department and the psychophysics laboratory of the Electronic Defense Group. The laboratory was headed by Wilson P. Tanner, Jr., a fellow graduate student in psychology but older and our mentor. During the early years at BBN, Green worked across the spectrum of psychoacoustics, manual control, educational technology, and pattern recognition. I worked in psychoacoustics, including an application to sound identification (such as in sonar) of computer-based instruction, and in pattern recognition. In 1962, when Licklider took a leave from BBN (described later), I obtained a leave from MIT and took over for him projects on computer-based instruction and computer-based libraries.

Green and I set up a computer-centered laboratory for signal detection research (see Figure 3.1) and obtained contract support to write a book on the topic.[21] Later, he was active in BBN's psychoacoustics department in the Los Angeles office while a professor at the University of California at San Diego, and then again in the Cambridge office while a professor at Harvard.[22] I stayed at BBN after my MIT leave expired and held several positions, including senior vice-president; general manager of research, development, and consulting; and member of the board of directors (all from 1970–74); and chief scientist for information sciences (1975–98).[23]

Licklider had a knack for attracting people to join his various endeavors:

> "Lick[lider] collected people," says his former student Tom Marill, who was struck by the way his mentor always tried to bring his favorites along as he moved from place to place. "He was very bright, he was very articulate, and because of that he was able to get very good people. They *liked* being collected."[24]

Not being able to leave it at that, I add that he was modest, generous, always in high

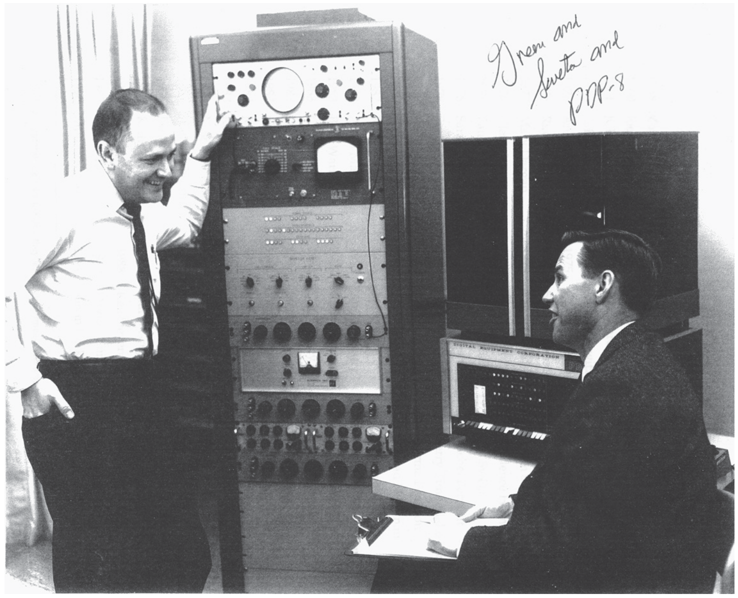


Figure 3.1. Dave Green and the author in their computer-centered laboratory for signal-detection research, in 1965. Further description of the laboratory and the role of its PDP-8 computer is given in Chapter 8. (Photo courtesy of BBN Technologies.)

spirits, with a sense of humor that made him fun to be around, and a very good friend. Talking with Licklider about a problem, according to Bill McGill, a Licklider colleague at Harvard and MIT, would amplify one's own intelligence by about 30 IQ points — a heady sensation.[25,26]

Others joining BBN at Licklider's urging included Richard Pew, an electrical engineer working in the Psychology Branch at Wright-Patterson AFB in Ohio, who joined BBN in 1958. He left shortly for a doctorate and then a professorship in psychology at the University of Michigan and returned to BBN in the early 70s. He pursued his specialty in human-computer interaction and headed the experimental psychology department (later, the cognitive science and systems department). Alexander Weisz joined BBN in 1960, researching pattern recognition and automated instruction for information-processing skills.[27]

Two more psychologists joined BBN in 1962. Licklider recruited engineering psychologist John Senders, formerly a student of his in a statistics course at Harvard, from a Honeywell human-factors laboratory. Senders managed an engineering psychology department at BBN and worked on manual control and tracking, pilots' eye movements, and visual sampling behavior and attentional demands in automobile driving.[28] Alfred Kristofferson, a friend of mine in the graduate program at Michigan, moved from the University of Cincinnati and pursued his research program on human timing capabilities along with studies of attention and computer-based learning.[29]

Licklider himself undertook psychoacoustic research, including the design of cockpit warning systems and the suppression of pain by music and white noise.[30] He advanced his ideas on human-machine integration and made an analysis of military pattern-recognition problems.[31] However, he concentrated on computers under company support and a contract from the Council on Library Resources, which was founded by the Ford Foundation to study "libraries of the future." The Council had consulted a dozen national leaders in related fields and converged on him to direct their study. The project began in late 1961 and concluded after two years. Licklider spent the second year on leave from BBN, but wrote the final report in 1963. This report was published as the book *Libraries of the Future*; it gives a prescient view of how future computer systems he termed "procognitive" could facilitate the acquisition, organization, and use of knowledge.[32]

3.6 From "B" to "C" at BBN

Why should BBN, or any organization, attempt to move to C from B — to computers from behavioral science or psychology? Consider the following factors. Psychologists interested in communications had studied information processing. They thought of computers as symbol processors — e.g., theorem provers and pattern recognizers — rather than as number crunchers. They would use computers to model human cognitive processes — dynamically rather than as previously via static mathematical equations — and would lend what they knew about human perception, thinking, language, and motor control to the design of computers that would augment or supplant human behavior, for example, in libraries, process control, and robotics. Psychologists had in their province the study of human and animal intelligence. They would contribute to automated speech recognition and to other instances of pattern recognition. Computers would be the prime case of a need for human-machine integration; they had very far to go in human-factors considerations to reach a semblance of user friendliness. The seminal idea of human-computer "symbiosis" — suggesting how the two could work together in complementary fashion — was forming in Licklider's thinking.[33]

3.7 Computers and time-sharing at BBN

Individuals arriving at BBN in the late 1950s to work on computers included Edward Fredkin in 1958, a computer scientist/engineer from Lincoln Laboratory where he had collaborated with Marill. Indeed, it was an LGP-30 computer — which Fredkin had ordered a bit earlier, before Licklider attracted him and when he thought he was going into business for himself — that BBN agreed to buy as part of the hiring arrangement.

After a mostly unsuccessful experience with that computer, Licklider jumped at the chance, in 1959, to have the Digital Equipment Corporation's (DEC's) prototype PDP-1 on the BBN premises (see Figure 3.2). This computer (called a Programmed Data Processor because the military was not buying "computers" at the time) stemmed from the Whirlwind, SAGE, and TX-0 developments at Lincoln Laboratory familiar to Licklider, Marill, and Fredkin. The PDP-1 had a "thin skin," meaning that it permitted an individual user to have convenient access via typewriter, punched tape, display screen, and light pen. (Not that it was easy to use: For example, two long rows of toggle switches, 35 in all, were used with the octal number system to check and change the contents of computer registers.) By 1959, apparently, the setting at BBN was one that DEC founder Kenneth Olsen could recognize as an appropriate test site for the PDP-1.

Fredkin, like Marill, had a large impact on BBN's assimilation and exploitation of the PDP-1, as described by Walden elsewhere in this volume (Chapter 4). Notably, he worked with DEC to specify the hardware changes that would be required to make possible "time-sharing" of the computer among multiple users. To emphasize the potential for the PDP-1 to interact with its environment, he programmed it to cut its own yellow ribbon at a ceremony held when the first production model was installed at BBN, in 1960.

Figure 3.2. Jerry Elkind, research assistant Donna L. (Lucy) Darley, and BBN's first PDP-1 computer, in 1960. (Photo courtesy of BBN Technologies.)

The time-sharing development at BBN, given a public demonstration in 1962, was the company's major computer project after the second PDP-1's arrival. It was led by Licklider in parallel with similar developments at MIT by professors John McCarthy and Marvin Minsky who consulted for BBN. Fredkin and Sheldon Boilen contributed ideas at BBN; Boilen and William Mann did most of the implementation.

Time-sharing a single computer among several users would have a significant economic impact, but from the historical vantage point, its major impact would come from allowing a user to be on-line and interactive with the computer in real time from his/her own terminal — in sharp contrast to submitting a stack of punched cards and, hours later, getting back a printout (assuming that the stack hadn't been inadvertently dropped or contained a typo). Users could now watch computers operate and begin to think about working with them cooperatively. Time-sharing was a major advance in human-computer integration and a sea change in the culture of computers and their users. It also made feasible the connection of large computers and multiple terminals in networks.

3.8 Licklider moves to ARPA

So by the summer of 1962, Licklider had published "Man-Computer Symbiosis," had spent three years interacting intensively with a PDP-1, and had initiated several computer projects, chief among them time-sharing and library function. At that point, he was recruited by the Advanced Research Projects Agency (ARPA) of the Department of Defense to manage two new offices: Information Processing Techniques and Behavioral Sciences. He accepted the position after convincing DOD officials that the future contributions of computers and humans to military command, control, and communication functions would best be served by his pursuing his interests in time-sharing and symbiosis, with generous support for academic research.

Others have written about how his choice of researchers, and of universities and a few other organizations as "centers of excellence," had a profound influence on the development of computer science in this country.[34] His chosen areas of research included time-sharing, artificial intelligence, speech recognition, natural language understanding, graphics, and visual pattern recognition, among others. A major project at MIT, to mention just one, was Project MAC, with initials connoting "machine-aided cognition" or "multiple-access computer." His ideas about an "inter-galactic network,"[35] realized later in the ARPANET, had a monumental impact, including on BBN (see Chapter 17 by Blumenthal et al.).

After his one-year stint at ARPA turned into two, Licklider did not return to BBN, but rather signed on with IBM. Three years there convinced him that IBM was not the place to develop his interests and he returned to MIT as a visiting, and then tenured, professor of electrical engineering.

The loss of Licklider hurt BBN doubly — not only form the loss of his intellect and skills but also financially, because Licklider had felt prohibited from supporting research at BBN in his ARPA role. However, BBN did receive support under his ARPA successors Ivan Sutherland, of the Information Processing Techniques Office (IPTO), and Lee Huff, of the Behavioral Sciences Office. Sutherland had gone to this position from Lincoln Laboratory, where he had done innovative work on computer graphics. His successor, Robert Taylor, initiated ARPA's support of the ARPANET. In his earlier position at NASA Headquarters, Taylor supported two BBN projects mentioned above: Elkind's work on manual control and Green and Swets's book on signal detection theory.[21] He had become acquainted with the work of these investigators through his graduate studies (at the University of Texas) in engineering psychology and psychoacoustics.[36]

The next head of IPTO was Lawrence Roberts, who was recruited specifically to provide technical and organizational leadership of the ARPANET project. Larry was once a student in my Psych 1 class at MIT, and more to the point, a staff member at Lincoln Laboratory where he collaborated with Marill (by that time head of his own company[37]in preliminary work on computer networking under ARPA support.[38] He knew Frank Heart at Lincoln and accepted BBN's proposal, spearheaded by Heart, to engineer and build the ARPANET. Following Roberts as IPTO head was Robert Kahn, earlier on the ARPANET project staff at BBN.[39] Kahn and IPTO contractor Vinton Cerf later developed the protocols for the Internet (for which they were given the National Medal of Technology).

3.9 Psychology's influence on computers at BBN: 1958–1963

The remainder of this chapter illustrates how Licklider and his colleagues gave shape to BBN's approach to computers that continues even today, drawn from the perspective of psychology. I briefly discuss 15 or so BBN projects undertaken in the five years from 1958, a period effectively coinciding with Licklider's stay. Many other BBN computer projects during those years and since are treated elsewhere in this volume (Chapter 4), as are several other strands of psychological work initiated later (Chapter 8).

To anticipate computer developments at BBN, let's briefly review what the approach to computers from psychology meant. It meant

- making computers available to individual users and easy for them to use
- creating facilities for symbiotic interaction with human problem solvers, wherein each component contributed according to its natural capabilities
- giving computers human-like capabilities of perception, thinking, speech, and motor control
- developing systems for the organization and availability and use of stored knowledge
- developing systems for information handling in specific settings, as in military command/control and hospitals

Such functions would require, first, computer time-sharing and second, computer networking — both as demonstrated at BBN.

In Licklider's own words from his article on symbiosis:

> Man-computer symbiosis . . . will involve very close coupling between the human and the electronic member of the partnership. The main aims are 1) to let computers facilitate formulative thinking as they now facilitate the solution of formulated problems, and 2) to enable men and computer to cooperate in making decisions and controlling complex situations without inflexible dependence on predetermined programs. In the anticipated symbiotic partnership, men will set the goals, formulate the hypotheses, determine the criteria and perform the evaluations. Computer machines will do the routinizable work that must be done to prepare the way for insights and decisions in technical and scientific thinking.[40]

In a recent review of work on human-computer interaction, Dick Pew captured some of Licklider's specifics as follows:

> He laid out the technological advances required to achieve these goals — developments in (1) computer time-sharing, because use of one machine for one knowledge worker was not, at the time, cost-effective; (2) hardware memory requirements because he foresaw the need for the user to have access to large quantities of data and reference material, a virtual library at one's fingertips; (3) memory organization because serial search through a sequentially organized database was too time-consuming and inefficient; (4) programming languages because of the extreme mismatch of languages the computer could understand and those the human could understand; and (5) input-output equipment because he envisioned the time when input and output should match the 'flexibility and convenience of the pencil and doodle pad or the chalk and blackboard used in technical discussion' (Licklider, 1960, p. 9).[41]

3.10 Libraries of the future

Several studies were conducted and several computer programs were written at BBN to improve human-computer interaction, as mentioned, under the auspices of a project to re-think traditional libraries. Licklider's book, *Libraries of the Future*, must be read to be appreciated; here I merely give a few pointers to its contents. Part I, entitled "Man's Interaction with Recorded Knowledge," describes the computer-based, symbiotic, "procognitive" system that should replace books and libraries based on books. It specifies 25 criteria that such a system should meet; gives an extended, hypothetical example of the personal use of such a system; and outlines the steps — mainly advances in computer facilities — toward realization of the system. The requisite capabilities of human-computer interaction are detailed. An introductory chapter estimates the size of the body of recorded information (based on Senders' work[42]). Because a procognitive system must have this corpus or much of it in a processible memory, the chapter relates the estimate of its size to estimates of the computer's memory size and processing speed.

The preface to Part II, "Explorations in the Use of Computers in Library and Procognitive Systems," is quoted as follows.

> Part II introduces and summarizes briefly 13 elements of the program of exploration into the uses of computers that constituted the major part of the two-year study. Chapter 5 is a survey of syntactic analysis by computer. Chapter 6 deals with quantitative aspects of files and text that bear upon the feasibility and efficiency of computer processing of library information. Chapter 7 describes a promising method for evaluating retrieval systems. Chapter 8 contrasts document-retrieval with fact-retrieval and question-answering systems. Chapter 9 describes eight efforts to develop, test, and evaluate computer programs that perform, or techniques that facilitate, library and procognitive functions.

The programs and techniques implement many of the ideas from previous chapters. I describe some of these elements next.

Automated syntactic analysis

Automated syntactic analysis was viewed as a precursor to computer processing and "understanding" of natural-language text. Danny Bobrow, an MIT computer science graduate student, surveyed the work on the English language.[43] Computer-oriented linguists had made various efforts to implement some theory of grammar in a computer program in order to assign words of a sentence to grammatical categories (or "parts of speech") and give a diagrammatic representation of the grammatical structure of a sentence. As one example, Chomsky's transformational rules were useful for handling two expressions of the same idea having different grammatical diagrams, such as with active and passive voice or two single-clause sentences and a compound sentence. At the time, Bobrow worked part-time at BBN. He joined the company full-time in 1965 to manage a new artificial intelligence department and, later, a computer sciences division.

Quantitative aspects of files and text

Given that procognitive systems require the storage in processible form of large amounts of text, information theorist Mario Grignetti studied the amount of memory required to store library information, both in indexes and actual text. Indexes contain the names or numbers of documents in a collection and, for each number, a list of terms or descriptors that characterize the corresponding document according to some coordinate indexing system. Ideally, terms are encoded for economy of storage space and ease of decoding. Grignetti found a "combinational code" to be truly efficient and the shortest possible code.[44]

Regarding storage requirements for the direct encoding of text, Grignetti re-examined Shannon's estimate of 11.8 bits per word and calculated, by a slightly different method, an information measure of 9.8 bits per word. The improvement of 20% suggested that further search for an efficient and economical coding scheme could be worthwhile.[45] Later, Grignetti, with David Bjorkman and Theodore Strollo, produced a program and the hardware specifications for the optimal technique for Rome Air Development Center's CDC-1604 computer.[46]

Evaluation of information-retrieval systems

In response to a query at hand, a perfect retrieval system (human or computer) would select all of the relevant documents (facts, answers) in the collection and none of the irrelevant ones. In practice, any system will miss some relevant items and select some irrelevant ones. Moreover, it will reflect some balance between proportions of relevant and irrelevant items selected, understanding that selecting more “true positives” will bring along more “false positives” and that selecting fewer false positives will decrease true positives. The system may be thought of as assessing the degree of relevance, for a given query, of every item in the collection and setting some cut point on that scale that must be exceeded by an item for it to be selected.

As with many diagnostic systems, performance data for retrieval systems, with any particular cut point, will yield a 2 x 2 table of relevance and retrieval. A model of data analysis from signal detection theory provides a way to measure effectiveness or accuracy that is unaffected by the selection cutoff and gives a separate measure of where that point is set.[47] Under the library project, I examined 10 measures of effectiveness and efficiency that had been suggested at the time and found the other measures lacking relative to the detection-theory measures.[48]

Further work was done under a contract from ARPA managed by Bobrow.[49] I then examined the performance data of three retrieval systems that had undergone extensive testing elsewhere, each with a particular collection of items. Two were computer-based systems and one was a manual system. Each was run with various retrieval methods, differing primarily with respect to how the query was framed, adding up to 50 methods. The results show small differences between methods for a given system/collection and substantial differences between systems/collections. Primarily, the results highlight the difficulty of the retrieval problem. In the best case found, retrieving on average 9 of 10 relevant items in a collection of 3000 would bring on average 300 irrelevant items mixed with them. To reduce the number of false positives to 30, say, by means of a strict cut point, one would receive only 4 of the 10 relevant items.[50]

Question-answering systems

As an alternative to a library system that provides documents, Marill analysed one that provides information. Such a system would read and comprehend the documents themselves, not merely their index terms or descriptors, and be able to organize the information. If the information is available, this system would accept questions in natural English and give answers in natural English.[51] Marill cited as an example the “Baseball” system proposed by Bert Green, Alice Wolfe, Carol Chomsky, and K. R. Laughery at Lincoln Laboratory that operates on stored baseball scores to answer such questions as “Did the Red Sox beat the Yankees five times in July?”[52] The key idea is that of a semantic net, an extension and formalization of the relational networks described by Licklider for procognitive systems.

Fischer Black, a mathematics graduate student at Harvard and part-time BBN employee, produced a series of question-answering systems that involved symbolic logic and computer programming, including one in which a non-human system first solved the “airport problem” posed by McCarthy. Based on some statements about a person’s whereabouts, transportation resources, and local geography, the system answers the question of how to get to the airport (in part: walk from my desk to my garage, drive my car to the airport).[53]

Associative chaining

A premise of question-answering systems is that an answer may have to be derived from elements of information scattered throughout the relevant body of literature. Information scientist Lewis Clapp devised computer programs to explore "chains" of relations between elements, which exist when they contain common words. Chains are of different orders corresponding to the number of intermediary items in a relation. His programs used graph theory to trace chains of relevance through bodies of literature consisting of files of sentences, and were thought to find possible application to the sets of descriptive terms used in coordinate-indexing systems.[54]

Symbiont

A "system to facilitate the study of documents," called "Symbiont," was designed and programmed by Licklider with three MIT graduate students working at BBN part-time: Danny Bobrow, Richard Kain, and Bert Raphael. It made available in an integrated package several functions useful in working with documents, such as retrieval of documents by abbreviated citations, designation and labeling of specific passages, Boolean search for desired passages, composition of graphs from tabulated data, and manipulation of graph coordinates and scales.[55] (See Figure 3.3.)

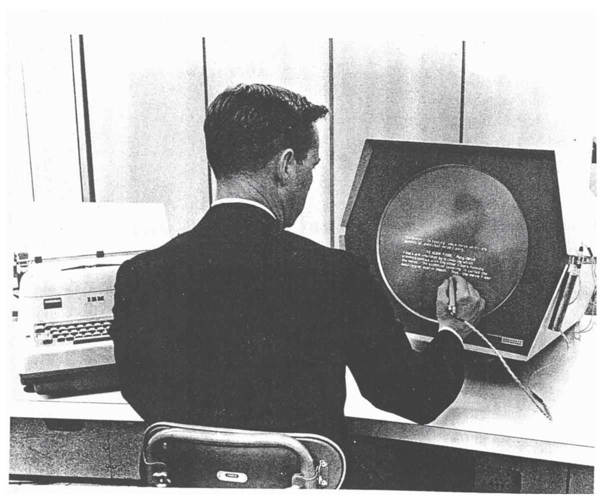


Figure 3.3. The present author working with the Symbiont system designed to facilitate the study of documents, in 1963. (Photo courtesy of BBN Technologies.)

Some utilitarian programs

Several computer programs at BBN were written merely to make it convenient to carry out some of the functions that are required in research on library and procognitive systems or in the efficient use of large collections of documents. For example, Licklider and engineer Welden Clark wrote an executive program to simplify and regularize the calling and returning of subroutines, to systematize the display of alphanumeric characters on typewriter and screen, and to display what the computer is doing. Specific to the last function were two programs jointly called "Introspection": "Program Graph" and "Memory Course" gave both a global view and considerable detail as to what was happening in the processor and memory of the computer — in preference to peeking at the contents of one register at a time.[56]

A direct file is ordered with respect to the items in the file and several descriptive terms are associated with each item. An inverse file is ordered with respect to its terms and several items are associated with each term. Grignetti's "file inverter" program operated on either the direct or inverse file to produce the other. Parts of that program were used to prepare an "automated card catalogue," which offered a user at a computer's typewriter several conveniences while attempting to retrieve relevant items.[57]

3.11 Program simplification

Tom Marill and BBN computer scientist T. G. Evans studied techniques of program simplification: one technique employed computational chains and the other, transformation rules.[58] Quoting from a later report:

> Experience has shown that if one can write one computer program which will find solutions to a given problem, then one can write several. . . . Optimal programming has the task of finding the best (in one sense or another) solution program for a given problem; e.g., a fastest program to invert matrices, or if one has a small computer, a program using the least possible amount of storage space, and among these, the shortest, and then the fastest. . . . However, at the present time, there is no formal theory of optimal programming. . . .The research reported here may be regarded as an attack on the problem of producing a theory. . . .What may be reasonably required of such a theory is that: (1) it produces optimizing programs so that the job of improving a given solution program can be left to the computer; and (2) it provides methods of judging whether or not a given optimizing technique preserves equivalence of programs . . . so that one can be sure that the improved program does the same thing as the original one.[59]

3.12 Automatic pattern recognition

Marill and Green examined an extension of signal detection theory as a model for a pattern recognizer consisting of a "receptor," which generates a set of tests of the physical sample to be recognized, and a "categorizer," which assigns each set of tests to one of a finite set of categories. Their first article focussed on rules of operation of the categorizer and how to optimize it. They went on to analyze how the effectiveness of a set of tests may be formally evaluated, without empirical study.[60]

Marill — with colleagues Alice K. Hartley, T. G. Evans, Burton H. Bloom, D. M. R. Park, Thomas P. Hart, and Donna L. Darley — produced a computer-based recognition system called Cyclops-1. The system recognized hand-printed alphanumeric characters, of different sizes and orientations, embedded in arbitrary numbers of them, overlapping or inside one another, superimposed on arbitrary backgrounds of meaningless lines, spots, or shapes. Items beyond alphanumeric characters could be added to the repertoire of items to be recognized, without affecting the recognition of items already in the repertoire.[61]

Warren Teitelman developed other methods for real-time recognition of hand-drawn characters (submitted for a Master of Science degree at MIT), his basic innovation being the use of time-sequence information (as used later prominently in speech recognition). A second innovation was the program's ability to modify its own performance by growing discrimination nets based on its experience. Moreover, the program could generate new tests dynamically, by having the human help it to learn to distinguish between two very similar characters that were previously identical for the program. Teitelman was a graduate student with Bobrow at MIT and later joined his department at BBN full-time.[62]

3.13 Computer-based teaching and learning

BBN's activities in computer-based learning were far-ranging, as the five areas described here illustrate.

Rote learning

Licklider wrote a teaching program for the PDP-1 to demonstrate how the computer could be an interactive teaching machine in the drill-and-reinforcement mode popularized by B. F. Skinner — and to encourage his children, Tracy and Linda, to learn German vocabulary. The results were published as a book chapter, based on a presentation at the Conference on the Application of Digital Computers to Automated Instruction, October 10–12, 1961.[63]

An engaging description of the teaching program, and Licklider's approach to the computer, comes from Ray Nickerson's unpublished "Reminiscences of BBN" [personal communication, Feb. 2003]. I quote from a passage in which Ray is writing about a few weeks he spent as a visitor to BBN, shortly before he joined the company for a long career.

> There are a few memories from this time that are vivid. One is of J. C. R. Licklider, 'Lick' to those who knew him — and one only had to meet him once to feel that one knew him — coke in hand, wheeling a file cabinet full of fan-fold paper tape holding his PDP-1 programs into the computer room, ready to start a hands-on-session with the machine. To us, as I suspect, to everyone who knew him, Lick projected a sense of enthusiasm and intellectual intensity that was almost palpable. He was a thinker and a visionary, but my impression was that he got enormous pleasure out of pushing bits around in his one-on-one sessions with the machine; and he was always eager to share his thinking and programming activities with anyone who showed an interest. (I remember Lick's back-up system. He had 7 trash barrels, each one marked with a day of the week. The rule was that the trash in each barrel — mostly discarded punched paper tape — was to stay around for a week before being dumped, so one had a week's grace period to retrieve any tape that had erroneously been discarded.)
>
> One of the programs that I encountered at BBN in those days that I remember particularly well was designed to help one learn lists of paired associates (states and capitols, presidents and their terms of office, English and foreign word equivalents). I think I learned of the program from Lick, but I did not know at the time that he had written it. It was simple in concept. On each 'trial,' it presented one of the items of a pair and the user had to type the corresponding item. The program was structured so that in order to get rid of an item — to not have the computer present it again — the user had to show some evidence that it had been learned. If one got an item correct the first time it occurred, it would not be presented again, but if one got it wrong on its on first occurrence, it would. Moreover, the more times one got an item wrong, the more times one would have to get it right in order to get rid of it. This meant that the computer quickly made one focus on those items one was having difficulty learning.
>
> The program had a number of 'bells and whistles' to give the learning session a bit of the feeling of playing a game. The computer made remarks that were appropriate to the level of learning efficiency the user was showing, and it gave a running score of how well (or poorly) one was doing in a given session. These features could be disabled by the flick of a switch, if one found them distracting or not wanted. It was a far cry from what is available today, but for its time it was a clever and innovative learning tool. (I used it to study German vocabulary in preparation for the language-qualifying exams that were required in my Ph. D. program, and found it to be quite effective). Lick and others at BBN went on to

> write considerably more sophisticated computer-assisted learning programs that made use of graphics to let one see immediately the effects of various operations on mathematical functions. More importantly, they clearly saw the potential of this technology for education and passed this vision on to others who helped develop this area of computer applications.

Student-controlled exploration

Just mentioned, but deserving its own heading, is Licklider's program to teach relations between symbolic and graphical representations of mathematical functions. A student typed the coefficients of a displayed equation (say, for a linear, parabolic, or square-root function) and the computer displayed the corresponding curve; the student could then vary the equation's coefficients to gain an intuitive understanding of the function. Another graphical program helped explore fundamentals of slopes and intercepts, with motion in the display to attract attention.

These modules were prepared under the contract mentioned above. In the book chapter, the time-sharing facility under development was given prominent treatment in an analysis of the economic aspects of computer-aided teaching.

Learning to identify nonverbal sounds

I submitted an unsolicited proposal to US Naval Training Device Center to examine various methods of instruction to determine how efficiently subjects could learn to identify a large number of nonverbal sounds, such as occur in sonar.[64] In these experiments the sounds consisted of five dispersed values along each of five dimensions, the dimensions being frequency, amplitude, interruption rate, duty cycle, and duration (3125 sounds in all). George Miller had reviewed data showing that an average of about seven one-dimensional stimuli could be correctly identified — the "magical number seven" — corresponding to transmission of about 2.6 bits of information.[65] He reported that adding dimensions helped, but less than expected; Pollack and Ficks used six dimensions and found that about 150 stimuli could be correctly identified, or 7.2 bits transmitted.[66]

Lincoln Laboratory psychologist Bert Green captured the essence of the project's experimental method in his book on *Digital Computers in Research*:

> The stimulus-generating ability of computers has been combined with the control possibilities by Swets (1961 [personal communication]), who programmed a computer to run an experiment in auditory recognition. The computer generates the complex auditory patterns that the subject is to identify, using the techniques described earlier in Sec. 10-3. The subject is seated before a typewriter connected directly with the computer. The computer types a message indicating that the experiment is about to begin and listing the symbols that the subject is to use to identify the various stimuli. Then a particular stimulus is presented. The subject makes a guess as to which stimulus it was and types the appropriate symbol. If he is correct, the computer proceeds to generate the next stimulus. If he is wrong, however, the computer displays the stimulus corresponding to the subject's choice and repeats the stimulus presented on that trial. The subject can compare the two and see his error and must then guess again. The computer is so fast and versatile that two subjects can be run in this experiment at the same time. The two subjects will be making different responses and thus will be running asynchronously. The computer can manage to keep them both occupied at the usual rates of responding.[67,68]

The instructional methods that I compared reflected various combinations of the principles of Skinner's automated instruction: continual interrogation and overt response; immediate knowledge of results; learner-controlled pacing of the lesson; and presentation of successive items conditional upon previous performance.[69] In brief, none of the methods was better than (in fact, as good as) a simple pairing of a stimulus and its corresponding symbol without overt response. Bits transmitted were no better than the Pollack and Ficks result of 7.2.[70]

Additional experiments were conducted to give the automated-instruction procedures another try, with enhanced methods. One enhancement gave the subject more control over the lesson, principally to be able to choose any of several methods at any time and to listen at will to various subsets of the total set of stimuli. Another intended improvement substituted, for the subject's typing a value (from 1 to 5) for each of five dimensions, a graphical display with light-pen. The subject could respond by pointing to five intersections of a displayed 5 x 5 matrix that represented spatially the sounds' dimensions (horizontally) and values (vertically). For feedback, the subject could compare the x's left by his/her pointer with the o's signifying the correct identification, to see easily the direction and extent of errors. The result in a word: accuracy of subject performance was no better than in the first experiment.[71,72]

Socratic instruction

In 1959, I believe, I wrote in a memo for a few colleagues a rather fanciful dialogue between medical student and computer to suggest how the two might interact in a Socratic manner to teach/learn an appropriate diagnostic procedure for a given patient history and set of symptoms. This memo was published by Wallace Feurzeig along with an article of his own after he had designed and programmed the "Socratic System."[73]

Feurzeig had worked at the Argonne National Laboratory in Illinois while it built computers number 11, 12 and 13 in the Eniac series and then he headed a computer group at the Laboratories for Applied Science in Chicago. He visited BBN in 1962 at John McCarthy's suggestion, attracted by BBN's capabilities in interactive computing. I, for one, lobbied department manager Tom Marill to make him a job offer. I proposed that he spend part of his time on developing a Socratic system, under the Wright-Patterson contract inherited from Licklider.

Tom agreed, and was kind enough to design a simple Socratic system to bring the concept along — one controlling a conversation between student and computer based only on logical conditions of sufficiency, necessity, redundancy, and consistency. His illustrative dialogue is no longer available, and so I reproduce one here from a similar system programmed by Judith Harris. It showed, we thought, that such a system could be of some interest, and possibly adequate for helping to teach subjects such as geometry or qualitative inorganic chemistry.

Under Harris's system, a student was to determine which letter of the alphabet the computer had selected as the target on a particular trial. The questions he/she could ask of the computer pertained to geometric characteristics of printed letters. In Figure 3.4 the student's questions and assertions are in capital letters; the computer's comments are in upper and lower case.

The Socratic System designed by Feurzeig was an early "intelligent tutor," which supported a more elaborate interaction in the interest of helping to form complex concepts, for example, in management decision-making or clinical medicine. For an illustrative medical problem — diagnosing pneumococcal pneumonia — the student's vocabulary (printed on a page) contained 40 questions (about the physical exam results or laboratory reports) and 35 declarative statements. The computer would respond to any of them with an answer, a comment, or, possibly, a question. The computer, as well as the student, could take the initiative. Its contributions depended on the

```
CURVES?

One.

LOOSE ENDS?

Two.

OBLIQUES?

Zero.

C.

You don't have enough information yet to get the right
answer.  How do you know it isn't J, for example?

J.

You don't have enough information yet.  How do you know it
isn't C, for example?

HORIZONTALS?

Zero.  That was a wasted question.  It did not eliminate
any possibilities.

VERTICALS?

Zero.  Good question.

G.

Wrong.  Your guess is inconsistent with the information you've been given.
You must have forgotten that the right letter hasn't any horizontal lines.
The letter you guessed can't be right--it has too many loose ends.

CURVES?

The answer is still:  one.

X.

Wrong.  Your guess is inconsistent with the information you've been given.
I think you only made that guess to see what I would say.

C.

Yes, you've done it.  Good work.  If you want to play again, start right in.
```

Figure 3.4 A very simple system for teaching via Socratic dialog.

student's actions up to that time and on the information he/she had at that point, and could depend on the order of previous interchanges. The condition of the patient, and the computer's responses to a given question, could vary over time. A subject-matter specialist and computer programmer could devise conditional strategies so that the computer answered good questions, reproved hasty conclusions, acknowledged perceptive decisions, questioned the grounds of inference, suggested new approaches, and developed interesting contingencies to the appropriate depth.

The illustrative medical dialogue mentioned and a description of the initial system are in the public literature.[74] Design of the medical problem benefited from the consultation of Dr. Preston Munter of the Harvard University Health Center. Alfred Kristofferson designed electronic trouble-shooting problems, also under Wright-Patterson Air Force Base support. Myra Breen provided utility programming for the system and assisted in preparing the applications.

Extensions and refinements of this work are described by Feurzeig elsewhere in this volume (Chapter 13). That chapter describes also a host of other innovative applications of computer-based teaching and learning that he developed over his five decade career at BBN.[75]

Second-language learning

In the fine ARPA tradition, its program director for behavioral sciences, Lee Huff, visited BBN and invited me to submit a proposal for a behavioral-sciences project — and accepted my proposal to develop computer techniques for teaching a second language, including its pronunciation as well as syntax and semantics.[76]

For the syntax-and-semantics system, Jaime Carbonell, a BBN acoustician in the process of turning cognitive scientist, and Mary Klatt, a linguist from the University of Michigan hired for the project, designed a computer interaction with a student via typewriter that could be used in either a teaching or testing mode. The interaction was in a conversational style, predominantly in the target language, and with the content and duration of the examination or lesson dependent upon immediate past performance.[77]

The major effort was a phonetics system, begun by Dennis Klatt, a speech scientist part-time from MIT, and Douglas Dodds, a BBN programmer. It performed an acoustic analysis of a student's utterance in real time and displayed visually any serious discrepancy between that utterance and its desired form in a way that indicated the changed articulation required for improvement. For example, tongue position for vowels was inferred from the frequencies of the first and second formants of speech, and an oscilloscope displayed schematically the trajectory of the student's tongue during production of a vowel along with the template trajectory for that student required for acceptable pronunciation. Consonant production and prosody (e.g., stress) were handled in a similar fashion.[78]

Linguists Bruce Fraser and Mary Klatt made an inventory of phonetic difficulties encountered in learning a second language. Richard Carter, like Fraser part-time from MIT, developed an approach to a theory of such phonetic difficulties. Ken Stevens and Mary Klatt made analyses of specific problems for vowels in going from Spanish to American English, and an analysis to quantify the acoustic differences between the two vowel systems.[79]

Daniel Kalikow, a BBN psychologist, served as principal investigator on the project in its final years. An article that he and I wrote presents an evaluation of (Spanish to English) displays for tongue location and trajectory during vowels, for isolating the vowels in multisyllabic words that should be reduced, and for the amount of aspiration of initial consonants and the time lapse before voicing of the succeeding vowel. Other displays were developed (e.g., for pitch) for teaching a tone language (Mandarin Chinese) to English speakers. Ann Rollins, Barbara Freeman, and Juan Anguita worked with us and Ray Nickerson, Ken Stevens, and Victor Zue (at MIT) advised. Experiments were conducted with Spanish-speaking Cambridge housewives, English-speaking students of Mandarin at two nearby universities, and with students in the Intensive English Program at the University of Miami.[80]

We worked with ARPA program officers to convince officials at three DOD language schools to give the system a try, without success; their programs were too intensive to leave room for experiments. An adaptation of the system was made later by Nickerson and colleagues, including Dan Kalikow and Ken Stevens (see Chapter 8 in this volume), seeking to improve the speech of children deaf from birth.[81]

3.14 A computer-based psychology laboratory

The “sound-learning “ project, as suggested above, provided the opportunity to develop what Nickerson and I think was the first computer-based laboratory for experiments in perception and learning. Mayzner and Goodwin, in the book *Minicomputers in Sensory and Information-Processing Research*, explained:

“Swets, Green, and [BBN research assistant Earl] Winter (1961), in a highly innovative pioneering effort, were developing one of the first truly automated minicomputer labs for the study of auditory discrimination and auditory information processing, and which was to become the *prototype* for almost all computer-automated auditory labs developed thereafter. Here for the first time a digital computer was being employed to generate auditory stimuli, compute their presentation sequence, feed back information to the subject concerning his responses, and analyze results, all in an interactive, real-time mode of operation.”[82,83]

3.15 Conclusion

The research and development firm known as BBN sought to enhance its physical and architectural acoustics activities by adding psychological acoustics in the mid 1950s. It hired psychologists to start that effort — for example, in speech and hearing — and to begin also a new activity in man-machine integration. The company’s capabilities then in communications, information processing, and man-machine integration suggested further an involvement with computers, at a time when preliminary developments, largely at MIT’s Lincoln Laboratory, indicated that computers could be more accessible, and hence more useful, to their primary users.

J. C. R. Licklider was hired as the key figure with the appropriate background and, especially, with the ideas and zeal to bring psychology to computers and computers to people; he began at BBN in 1957 and stayed until 1962. A computer derived from Lincoln’s computers and able to support his ideas, namely, DEC’s PDP-1, soon became available to him at BBN.

Licklider wanted computers to be directly available to individual workers in knowledge fields, to help these users go about their intellectual work, and to be understandable to them. He also wanted computers to help people learn a variety of skills. He had a good sense of what computers and humans could do, complementarily, in cooperation.

To accomplish these aims, BBN hired over the next few years about a dozen computer scientist/ engineers, mostly from MIT and Lincoln, and as many experimental/engineering psychologists, most of whom had worked with Licklider before. Each of them was involved in a range of research and development projects — with government, private, and company support.

When he left BBN, he infused computer science across the country with these same programmatic ideas from his "pulpit," and with his resources, at ARPA. His successors there became aware that the team he built at BBN was capable of helping to carry them out. BBN participated in ARPA's programs in artificial intelligence, speech recognition, natural language understanding, and intelligent tutors, among others. The major ARPA-BBN project, of course, was computer networking — the icing on the cake for human-computer symbiosis. And networking brought us a new form of human-human interaction — thanks to BBNer Ray Tomlinson's creation of email.

In 1975, to take one snapshot from detailed descriptions elsewhere in this volume, BBN had departments named artificial intelligence, control systems, distributed information systems, educational technology, experimental psychology, interactive systems, psychoacoustics, sensor signal processing, and speech signal processing — in an Information Sciences Division directed by Ray Nickerson. Meanwhile, Frank Heart directed a Computer Systems Division, with major activities in networking and life sciences. Together, these groups had a staff well on its way to the BBN peak of 500 or so computer and cognitive-science professionals in its research and development activities, with upwards of a hundred projects active at any time. It is not too great a stretch to say that most of these staff members were carrying out projects in a conceptual line from BBN's early computer themes — brought from A to B to C.

Acknowledgments

My thanks to BBN alumni Dave Walden and Ray Nickerson for wise editorial guidance, to Mary Parlee at MIT for sharing knowledge of the history of psychological science and of writing recent science history, and to Jennie Connolly for always being ready to delve into the BBN library. Dick Pew and Duncan Luce kindly read a draft and refined some of my recollections.

References and Notes

1. I have been asked by the editors to write about people and their connections as well as technical subjects and to include some personal data. Doing so comes naturally because the course of events I describe depended substantially on who knew whom when, and some characterization of my vantage point may also be relevant. For the most part, such material will be confined to numbered notes. (These notes give a full citation for a literature reference the first or only time it occurs and a reduced citation thereafter.)

2. P. N. Edwards, *The Closed World: Computers and the Politics of Discourse in Cold War America*, MIT Press, Cambridge, MA, 1996; K. Hafner and M Lyon, *Where Wizards Stay Up Late: The Origins of the Internet*, Simon and Schuster, New York, 1966; T. P. Hughes, *Rescuing Prometheus*, Vantage, New York, 1998; M. M. Waldrop, *The Dream Machine: J. C. R. Licklider and the Revolution That Made Computing Personal*, Viking, New York, 2001.

3. C. E. Shannon, "The Mathematical Theory of Communication," *Bell System Tech. J.*, vol. 27, July 1948, pp. 379–423, and Oct. 1948, pp. 623–656.

4. P. N. Edwards, *The Closed World*, Table 7.1, p. 215.

5. J. C. R. Licklider, "Basic Correlates of the Auditory Stimulus," *Handbook of Experimental Psychology*, S. S. Stevens, ed., Wiley, New York, 1951, pp. 985–1039; J. C. R. Licklider and G. A. Miller, "The Perception of Speech," pp. 1040–1074; G. A. Miller, "Speech and Language," pp. 789–810. The breadth of these psychologists, beyond the PAL, is suggested by their articles on hoarding behavior in the white rat.

6. Licklider's career is described by the historians cited in note 2 above and by R. M. Fano, "Joseph Carl Robnett Licklider," March 11, 1915–June 26, 1990, *Biographical Memoirs*, National

Academy of Sciences, National Academy Press, Washington, D.C., vol. 75, pp. 191–214. Miller's accomplishments are described by historians Bernard J. Baars, *The Cognitive Revolution in Psychology*, Guilford, New York, 1986, and P. N. Edwards, *The Closed World*, and by his colleagues in William Hirst, ed., *The Making of Cognitive Psychology: Essays in Honor of George A. Miller*, Cambridge University Press, New York, 1988.

7. See, e.g., J. C. R. Licklider, "Three Auditory Theories," *Psychology: A Study of a Science (Sensory, Perceptual, and Physiological Formulations)*, S. Koch, ed., McGraw-Hill, New York, 1959, pp. 41–144 and "The System System," *Human Factors in Technology*, E. Bennett, J. Degan, and J. Spiegel, eds., McGraw-Hill, New York, 1963, pp. 627–641.

8. G. A. Miller, "What Is Information Measurement?," *Am. Psychologist*, vol. 8, no. 1, Jan. 1953, pp. 3–11; Ibid., "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," *Psychological Rev.*, vol. 63, no. 2, Mar. 1956, pp. 81–97; N. Chomsky and G. A. Miller, "Finite State Languages," *Information and Control*, vol. 1, no. 2, 1958, pp. 91–112. Three chapters by the last two authors appeared in the *Handbook of Mathematical Psychology, Vol. II*, R. D. Luce, R. R. Bush, and E. Galanter, eds., Wiley, New York, 1963.

9. P. N. Edwards, *The Closed World*, p. 179.

10. The proceedings of the second conference were published in *J. Acoust. Soc. Am.*, vol. 22, no. 6, Nov. 1950, pp. 689–806. Mary Parlee, who is writing a history of psychology and brain science at MIT, told me of these conferences.

11. The conference proceedings were published in *IRE Trans. Info. Theory*, Prof. Group Info. Theory, vol. IT-2, no. 3, Sept. 1956. On the conference's anticipation of cognitive science, see, e.g., B. J. Baars, *The Cognitive Revolution in Psychology*, and P. N. Edwards, *The Closed World*.

12. G. A. Miller, E. Galanter, and K. H. Pribram, *Plans and the Structure of Behavior*, Holt, New York, 1960.

13. I used Miller's book *Language and Communication* (McGraw-Hill, New York, 1951) as a text in courses I taught at Michigan and MIT. He included an article based on my doctoral thesis (J. A. Swets, W. P. Tanner, Jr., and T. G. Birdsall, "Decision Processes in Perception," *Psychological Rev.*, vol. 68, no. 5, Sept. 1961, pp. 301–340) in his collection of articles on *Mathematics and Psychology*, Wiley, New York, 1964, pp. 184–197.

14. As a comment on information models from today's perspective: "The idea of the mind being an information-processing network with capacity limitations has stayed with us, but in far more complex ways than pure information theory." Quoted from R. D. Luce, "Whatever Happened to Information Theory in Psychology?," *Rev. Gen. Psych.*, vol. 7, no. 2, 2003, pp. 183–188. Luce considers why initial enthusiasm in psychology for Shannon's quantification of information capacity has not been sustained.

15. W. A. Rosenblith, K. N. Stevens, and the Staff of BBN. *Handbook of Acoustic Noise Control: Volume II, Noise and Man*, tech. report WADC 52-204, Wright Air Development Center, 1953; K. N. Stevens, *A Survey of Background and Aircraft Noise in Communities Near Airports*, tech. report NASA 3379, National Aeronautics and Space Administration, 1954; K. N. Stevens, W. A. Rosenblith, and R. H. Bolt, "A Community's Reaction to Noise: Can It Be Forecast?," *Noise Control*, vol. 1, no. 1, Jan. 1955, pp. 63–71; K. N. Stevens and J. J. Baruch, "Community Noise and City Planning," chapter 35, *Handbook of Noise Control*, C. M. Harris, ed., McGraw-Hill, New York, 1957).

I won't generally list later positions and honors, but will identify the authors just listed by mentioning that Stevens won the National Medal of Science; Rosenblith, a PAL alumnus, became MIT provost; Bolt served as Associate Director of the National Science Foundation before joining BBN full-time as chair of its board; and Baruch, a BBN partner when the number of partners reached the ultimate five, later served as Assistant Secretary of Commerce for Science and Technology. (While the original version of this chapter was being completed, in late 2003, the National Medal of Science was awarded to Leo Beranek.)

16. Giving up psychology at MIT and Lincoln Laboratory was a tough decision for Licklider because he had assembled a "dream team"; see M. M. Waldrop, *The Dream Machine*, p. 105. Of that team, Fred Frick and Bill Harris remained at Lincoln through their careers and Jim Degan

moved to Lincoln offshoot The MITRE Corporation, but the others moved on: Bert Green to the Carnegie Institute of Technology, as department chair, and then to Johns Hopkins; Herb Jenkins to Bell Labs and (when the telephone company decided it didn't need a pigeon lab) to McMaster University; Bill McGill to Columbia and then to the University of California at San Diego, including a period as chancellor, and then to Columbia as president; George Miller in succession to Harvard, Rockefeller University, and Princeton; Keith Smith to Michigan, including a long term as department chair; Warren Torgerson to Johns Hopkins; Ben White to San Francisco State University, and Douwe Yntema to Harvard. (Smith, a friend and classmate of mine at Michigan, at one point remained at Lincoln rather than accept the faculty position at MIT that opened when McGill left, because he preferred "two birds in the hand to one in the bush." That gave me pause as I accepted the campus job rather than an offer from Lincoln.) One point of this note is to illustrate the observation by Mary Parlee that MIT psychologists moved frequently in and out between universities and settings for applied research.

These psychologists began a monthly evening meeting to talk about ideas, called the Pretzel Twist. It continued during my term at MIT with the next generation of MIT psychology faculty, Roger Brown, Dave Green, Davis Howes, Ron Melzack, and Michael Wallach, and some Harvard faculty: I recall Dick Herrnstein, Duncan Luce, and Roger Shepard. Most of these individuals met yearly for a weekend with an invited group in the East called the Psychological Round Table, where each attendee tried to give a talk about his work amid frequent interruptions for information, criticism, or humor. Ejected by policy from the PRT at age 40, with some attrition and creeping sedateness they continued to meet annually under the auspices of a national honorary society established in 1904, soon after and today called the Society of Experimental Psychologists.

17. Licklider's excitement about computers led some of his friends to think that he lost interest in psychology and psychoacoustics at this time. Bob Fano's memoir on him for the National Academy of Sciences has a different view:

"It is interesting to note that, while Lick[lider] was nominated for membership in the National Academy of Sciences by its Psychology section, upon his election in 1969 he chose the Engineering section as his home, having become by then a leading member of the computer sciences community. Yet, after walking through Lick's career in an attempt to understand the evolution of his intellectual interests and motivations, I came to the conclusion that he was first and foremost a psychologist throughout his professional life. Psychoacoustics was his primary, long-lasting interest and his source of motivation; his very last research project was still motivated by his interest in modeling psychoacoustic phenomena" (R. M. Fano, " Joseph Carl Robnett Licklider," p. 208).

18. Karl Kryter published 10 articles, and wrote upwards of 20 technical reports, from BBN over 8 years. Titles of the articles are: noise control criteria for buildings, the meaning and measurement of perceived noise level, some effects of spectral content and duration on perceived noise level, reaction of people to exterior aircraft noise, airports and jet noise, damage risk criterion and contours based on permanent and temporary hearing loss data, temporary threshold shifts in hearing from acoustic impulses of high intensities, acceptability of aircraft noise, study of the acoustic reflex in infantrymen, automatic evaluation of time-varying communications systems. Three principal technical reports are: K. D. Kryter and J. H. Ball, *An Evaluation of Speech Compression Techniques*, tech. report RADC-TDR-63-90, BBN, 1963 (supported by the Rome Air Development Center, Griffiss Air Force Base, New York); K. D. Kryter, *The Effects of Noise on Man*, BBN Report 1299, 1965 (supported by the U.S. Army Medical Research and Development Command, Office of the Surgeon General, Washington, D.C.); K. S. Pearsons and K. D. Kryter, *Laboratory Tests of Subjective Reactions to Sonic Boom*, tech. report NASA CR-187, BBN, 1965 (supported by the National Aeronautics and Space Administration).

19. Dewey Neff wrote three technical reports during three years at BBN: W. D. Neff, *Neural Mechanisms of Sensory Discrimination*, BBN Report 1126, 1963 (supported by the Physiological Psychology Branch, Psychological Sciences Division, Office of Naval Research, Department of the Navy, Washington, D.C., monitored by G. C. Tolhurst); *Neural Mechanisms for Responses of Middle Ear Muscles*, BBN Report 1128, 1963 (supported by the U.S. Army Research and Development Command, Office of the Surgeon General, Washington, D.C.; *Transmission and*

Coding of Information in Auditory Nervous System, BBN Report 1127, 1964 (supported by the Directorate of Life Sciences, Air Force Office of Scientific Research, Department of the Air Force, Washington, D.C.).

20. Vin Sharkey wrote five technical reports from BBN, four of them classified as secret.

21. D. M. Green and J. A. Swets, *Signal Detection Theory and Psychophysics*, Wiley, New York, 1966. Reprinted with corrections by Robert E. Krieger, Melbourne, Fl., 1974; now in print by Peninsula, Los Altos, Calif., 1988.

22. Green's appointment in the psychology department at Harvard was as "Professor of Psychophysics," a title created originally for Smitty Stevens, director of PAL.

23. I chose to stay at BBN because I liked its culture and people and appreciated the relief it offered from teaching and other academic duties as well as the opportunity to do both basic and applied research; I could continue to pursue pure psychophysics and also do something more humanitarian. There was, however, a push from MIT as well as a pull from BBN. The six psychology professors there did not enjoy life under a new chairman and left within a two-year period.

Anecdote, about Beranek. The story has been told of Leo's travelling to Los Angeles in the summer of 1956 to visit Licklider and his wife Louise — to convince them not to accept a job at Hughes Aircraft but to return to Cambridge so that she could continue her activities in drama and he could join BBN (M. M. Waldrop, *The Dream Machine*, p. 151). Leo involved me in a similar gambit a few years later. Specifically, after a year or so at BBN, I planned to interview for a faculty position at the University of Michigan while attending a meeting of the Acoustical Society in Ann Arbor. Leo must have heard of these plans because he suggested that we go to the meeting together and share a room at the Michigan Union. After each interview, he would run into me. I knew he had done his job when, after he took my parents to dinner, my mother said she would like to have my family nearby but could understand why I would like to work with such a fine man as Dr. Beranek.

24. M. M. Waldrop, *The Dream Machine*, p. 152.

25. I paraphrase McGill from M. M. Waldrop, *The Dream Machine*, p. 7.

26. As friend and patron, Licklider introduced Dave Green and me to the Acoustical Society and was instrumental in our elections as Fellows a few years later. Dave went on to win the Society's three major medals and serve as president. As president, he joined predecessors Bolt, Beranek, Licklider, and Kryter, as well as three or four other BBNers from the physical acoustics part of the company.

27. A. Z. Weisz, J. C. R. Licklider, J. A. Swets, and J. P. Wilson, *Human Pattern Recognition Procedures as Related to Military Recognition Problems*, BBN Report 939, 1962 (supported by the Electronics Research Directorate, Air Force Cambridge Research Laboratories, Office of Aerospace Research, L.G. Hanscom Field, Bedford, Mass.). A. Z. Weisz, *Automated Instruction Procedures for Training Information-Processing Skills: Recommendations for a Computer-Based Program of Research*, BBN Report 1122, 1963 (supported by the Decision Sciences Laboratory, Electronics System Division, Air Force Systems Command, L.G. Hanscom Field, Bedford, Mass.).

28. J. W. Senders, "Information Storage Requirements for the Contents of the World's Libraries," *Science*, vol. 141, no. 3585, Sept. 13, 1963, pp. 1067–1068; *An Investigation of Visual Sampling Behavior of Human Observers*, BBN Report 1335, 1963 (supported by NASA/Langley Research center, Langley Station, VA); *Multiple Criteria for Assessing Tracking Performance: The Assessment of Step Input Tracking*, BBN Report 1287, 1965 (supported by the Engineering Psychology Branch of the Office of Naval Research, U.S. Navy); J. W. Senders, A. B. Kristofferson, and W. Levison, *An Investigation of Automobile Driver Information Processing*, BBN Report 1246, 1966 (supported by the Bureau of Public Roads, Department of Commerce, Washington, D.C.).

29. A. B. Kristofferson, "Successiveness Discrimination as a Two-State Quantal Process," *Science*, vol. 158, no. 3806, 8, 1967, pp. 1337–1339; J. A. Swets and A. B. Kristofferson, "Attention," *Ann. Rev. Psych.*, vol. 21, 1970, pp. 339–366; see W. Feurzeig, J. R. Harris, J. A. Swets, *The Socratic System: A Computer System for Automated Instruction*, BBN Report 1065, 1963 (supported by the Behavioral Sciences Laboratory, 6570th Aerospace Medical Research Laboratories, Wright-Patterson Air Force Base, Dayton, Ohio).

30. J. C. R. Licklider, *Audio Warning Signals for Air Force Weapon Systems*, BBN Report 746, 1960 (supported by the Wright Air Development Division, Wright-Patterson Air Force Base).

The problem with cockpit warnings was that sometimes a loud klaxon to warn the pilot during landing that the wheels were up went unnoticed; there was more science to it, but I seem to recall that a recording of a woman's voice was effective.

The audio-analgesia work was reported in "On Psychophysiological Models," *Sensory Communication*, W. A. Rosenblith, ed., MIT Press, Cambridge, Mass., 1961, pp. 49–72.

Dr. Wallace J. Gardner, a local dentist, had found that sounds in earphones, controlled both by a dentist and a patient, were effective in suppressing pain for most patients during dental operations. Licklider developed a psychophysiological model for the process, aided by experiments on pain conducted by Alex Weisz and Ron Melzack, the latter an MIT psychologist working part time at BBN (I served Melzack as an experimental subject, transported by my favorite music as my bare foot remained in a pail of ice water).

31. *Studies in the Organization of Man-Machine Systems*, BBN Report 970, 1962 (supported by the Behavioral Sciences Division, Air Force Office of Scientific Research, Washington, D.C.). A. Z. Weisz, et al., *Human Pattern Recognition Procedures as Related to Military Recognition Problems*.

Dick Pew told me that Licklider gave him and Dave Green the Request for Proposal for the pattern-recognition project on a Tuesday with a mailing date for the proposal on the following Tuesday. Dave returned the RFP to Licklider on Friday with the comment that he and Dick found that they could not generate a good proposal. Licklider worked on it over the weekend and mailed it on Tuesday. When the soliciting agency's review was completed and the contract awarded to BBN, the responsible technical officer asked whether it was the proposal or the final report.

Licklider had a way of brilliantly recasting the statement of a problem in an RFP while making his enhancements seem like the full intentions of the statement's author. That author would realize that the contract could not possibly be let to anyone proposing to work on the pedestrian problem as originally conceived.

32. J. C. R. Licklider, *Libraries of the Future*, MIT Press, Cambridge, Mass., 1965. If memory serves, Licklider dictated the final report while giving a week's forced bed rest to a troublesome back at a San Diego motel. Project secretary Millie Webster and I in Cambridge were called frequently that week and the project staff was on alert.

33. J. C. R. Licklider, "Man-Computer Symbiosis," *IRE Trans. Human Factors in Eng.*, vol. 1, no. 1, Mar.1960, pp. 4–11.

34. In addition to the references in Note 2 above, see *Funding a Revolution: Government Support for Computing Research*, National Academy Press, Washington, D.C., 1999 and C. I. Kita, "J. C. R. Licklider's Vision for the IPTO," *IEEE Ann. Hist. Comp.*, July-Sept. 2003, pp. 62–77.

35. J. C. R. Licklider, "To Members of the Intergalactic Computer Network," ARPA Memorandum, 25 April 1963.

36. Taylor studied with Acoustical Society regular Lloyd Jeffress, a good friend of the psychoacousticians at BBN, who had organized the famed 1948 Hixon Symposium, published as L. A. Jeffress, ed., *Cerebral Mechanisms in Behavior*, Wiley, New York, 1951, which included chapters by John von Neumann (The General and Logical Theory of Automata), Warren McCulloch (Why the Mind is in the Head), Karl Lashley (The Problem of the Serial Order of Behavior) and Wolfgang Kohler (Relational Determination in Perception).

37. Tom Marill's company was CCA, the Computer Corporation of America. In 1974, he asked me to think about becoming its president for a new phase of its development. It was an invitation I treasure, along with an offer of a research-staff position from Jerry Elkind when he was with RCA.

38. T. Marill and L. G. Roberts, "Toward a Cooperative Network of Time-Shared Computers, *AFIPS Conf. Proc. Fall Joint Comp. Conf.*, Spartan Books, vol. 29, 1966, pp. 425–431.

39. Bob Kahn reminded me a few years ago that I gave him his first annual review at BBN. He remembered that when I asked him what he had accomplished and he said "Well, with mathematics it's sometimes hard to know," I replied that if he didn't know then I surely didn't and he would not get a pay raise. I don't recall if we left it at that.

40. J. C. R. Licklider, "Man-Computer Symbiosis," p. 4.

41. R.W. Pew, "Evolution of Human-Computer Interaction; from Memex to Bluetooth and Beyond," in *The Human-Computer Interaction Handbook: Fundamentals of Evolving Technologies and Emerging Applications*, J. A. Jacko and A. Sears, eds., Erlbaum, Mahwah, N.J. 2003, p. 4.

42. J.W. Senders, "Information Storage Requirements for the Contents of the World's Libraries," *Science*, vol. 141, no. 3585, 13 September 1963, pp. 1067–1068.

43. D. G. Bobrow, "Syntactic Analysis of English by Computer — A Survey," *Proc. Am. Fed. Info. Processing Societies*, vol. 24, Nov. 1963, pp. 365–387.

44. M. C. Grignetti, *On the Length of a Class of Serial Files*, BBN Report 1011, 1963.

45. M. C. Grignetti, "A Note on the Entropy of Words in Printed English," *Information and Control*, vol. 7, no. 3, Sept. 1964, pp. 304–306.

46. M. C. Grignetti, D. J. Bjorkgren, and T. Strollo, *Code Compression Program*, BBN Report 1247, 1965.

47. J. A. Swets, "Measuring the Accuracy of Diagnostic Systems," *Science*, vol. 240, no. 4857, 3 June 1988, pp. 1285–1293.

48. J. A. Swets, "Information Retrieval Systems," *Science*, vol. 141, no. 3577, 19 July 1963, pp. 245–250. Reprinted in M. Kochen, ed., *The Growth of Knowledge*, New York, Wiley, 1967, pp. 174–184; T. Saracevic, ed., *Introduction to Information Science*, New York, Bowker, 1970, pp. 576–583.

49. Supported by ARPA through the Air Force Cambridge Research Laboratories, Office of Aerospace Research, monitored by Stanley R. Petrick of the Decision Sciences Laboratory.

50. J. A. Swets, "Effectiveness of Information Retrieval Methods," *Am. Documentation*, vol. 20, no. 1, Jan. 1969, pp. 72–89. Reprinted in B. Griffith, ed., *Key Papers in Information Science*, Knowledge Industry Publications, White Plains, New York, 1980, 349–366.

51. T. Marill, *Libraries and Question-Answering Systems*, BBN Report 1071, 1963.

52. B. F. Green, A. K. Wolfe, C. Chomsky, and K. R. Laughery, "Baseball: An Automatic Question Answerer," *Proc. Western Joint Computer Conf* , vol. 19, 1961, pp. 219–224.

53. F. S. Black, *A Question-Answering System; QAS-5*, BBN Report 1063, 1963. Black later invented with M. S. Scholes and R. C. Merton a method for valuing stock market options that won the Nobel Prize for the other two collaborators after his death.

54. L. C. Clapp, *Associative Chaining as an Information-Retrieval Technique*, BBN Report 1079, 1963.

55. D. G. Bobrow, R. Y. Kain, B. Raphael, and J. C. R. Licklider, "A Computer-Program System to Facilitate the Study of Technical Documents," *Am. Documentation*, vol. 17, no. 4, Oct. 1966, pp. 186–189.

56. J. C. R. Licklider and W. E. Clark, "On-Line Man-Computer Communication," *Proc. Am. Fed. Info. Processing Societies*, vol. 21, May 1962, pp. 113–128.

57. M. C. Grignetti, *Computer Aids to Literature Searches,* BBN Report 1074, 1963.

58. T. Marill and T. G. Evans, *Computational Chains*, BBN Report 778, 1960; T. G. Evans and T. Marill, *Transformation Rules and Program Simplification*, tech report 784, BBN (supported by the Astrosurveillances Sciences Laboratory, Electronics Research Directorate, Air Force Cambridge Research Laboratories).

59. T. Marill, *Techniques of Simplification*, BBN Report 1006, p. 1.

60. T. Marill and D. M. Green, "Statistical Recognition Functions and the Design of Pattern Recognizers," *IRE Trans. Elec. Computers*, vol. EC-9, no. 4, Dec. 1960; "On the Effectiveness of Receptors in Recognition Systems," *IEEE Trans. Prof. Tech. Group Info. Theory*, vol. IT-9, no. 1, Jan. 1963, pp. 11–17 (supported by the Air Force Cambridge Research Laboratories).

61. T. Marill, A. K. Hartley, T. G. Evans, B. H. Bloom, D. M. R. Park, T. P. Hart, and D. L. Darley, "Cyclops-1: A Second-Generation Recognition System," *Proc. Amer. Fed. Info. Processing Societies, Fall Joint Computer Conf.,* vol.24, 1963, pp. 27–33 (supported by the Office of Aerospace

Research, Air Force Cambridge Research Laboratories); B. H. Bloom and T. Marill, *Cyclops-2: A Computer System that Learns to See*, BBN Report 1333, 1965 (supported by the Air Force Cambridge Research Laboratories).

62. W. Teitelman, *New Methods for Real-Time Recognition of Hand-Drawn Characters*, BBN Report 1015, 1963.

Jumping ahead a few years, Danny Bobrow and Warren Teitelman stayed at BBN until 1971 when they were recruited to Xerox PARC (Palo Alto Research Center) by the director of its new Computer Science Laboratory, Jerry Elkind. Jerry had been recruited by Bob Taylor. PARC's large contribution to the development of the personal computer is, of course, another whole story; see, e.g., M. M. Waldrop, *The Dream Machine*. To the large extent it came under Taylor's and Elkind's leadership, it harks back directly to Licklider's influence.

63. J. C. R. Licklider, "Preliminary Experiments in Computer-Aided Teaching," *Programmed Learning and Computer-Based Instruction*, J. E. Coulson, ed., Wiley, New York, 1962, pp. 217–239. The work was supported by the United States Air Force and was monitored by the Training Psychology Branch, Behavioral Sciences Division, Aero Space Medical Laboratory, Aeronautical Systems Division, Air Force Systems Command [at Wright-Patterson AFB, Dayton, Ohio]; Dr. Felix Kopstein and Dr. Theodore Cotterman were project monitors. I was principal investigator under the contract in its second year (1962–63) and Dr. Ross Morgan was project monitor. For the second year, the project work was converted to Socratic teaching, as described shortly.

64. Project monitored by Dr. James J. Regan, from 1959 to 1963.

65. G. A. Miller, "The Magical Number Seven, Plus or Minus Two."

66. I. Pollack and L. Ficks, "Information of Elementary Multi-Dimensional Auditory Displays," *J. Acoust. Soc. Am.*, vol. 26, no. 2, Mar. 1954, pp. 155–158.

67. B. F. Green, *Digital Computers in Research*, McGraw-Hill, New York, 1962, pp. 189–190.

68. Licklider, in "Preliminary Experiments in Computer-Aided Teaching," p. 237, identified the sound-learning project as the first application of BBN's time-sharing capability.

69. B. F. Skinner, "Teaching Machines," *Science*, vol. 128, no. 3330, Oct. 24, 1958, pp. 969–977.

70. J. A. Swets, S. H. Millman, W. E. Fletcher, and D. M. Green, "Learning to Identify Nonverbal Sounds: An Application of a Computer as a Teaching Machine," *J. Acoust. Soc. Am.*, vol. 34, no. 7, July 1962, pp. 928–935.

71. J. A. Swets, J. R. Harris, L. S. McElroy, and H. S. Rudloe, "Computer-Aided Instruction in Perceptual Identification," *Behavioral Sci*, vol. 11, no. 2, Mar. 1966, pp. 98–104.

72. Ed Fredkin suggested to me that the PDP-1 could generate the sounds and send them to earphones along a wire appropriated from the display scope — a far cry from the analogue possibilities I was discussing with Licklider (I think he suggested that I buy 3125 tape players). Ed wrote a (sine wave) program in a matter of hours that permitted me to produce any value along each sound dimension. Alan Tritter, who billed himself as the world's greatest programmer (he weighed in excess of 300 pounds), came by and offered to write a decimal-to-octal conversion so I could work in the more familiar number system. The computer frustrated him for 15 minutes at which time he came down on the teletype with a mammoth fist and dented it enough to make me find another. William Fletcher, a military jet pilot and Cal Tech undergraduate with Fredkin who had joined BBN, stayed up overnight to program the features that allowed versatile control of the lesson and data analysis. Harry Rudloe programmed the graphic displays and controls, with a general purpose in mind. The experiments were run by Earl Winter, Susan Millman, Judith Harris, and Linda McElroy. Judy had been a teaching assistant to me at MIT. Linda came from Lincoln Laboratory; she was house-sitting for the Lickliders in the fall of 1956 when my wife and I and our two preschool sons moved in for a few weeks upon arriving in Massachusetts from Michigan.

The preceding paragraph reminds me of Licklider's theory of least effort for administration, perhaps worthy of a single example. In September 1956, I had sold a house in Ann Arbor, bought a new car, driven my family to Massachusetts, and bought a house there when I went to the office of Ralph Freeman, chairman of MIT's Department of Economics and Social Sciences, to introduce myself as his newest assistant professor. He looked puzzled and said he didn't recall

Licklider telling him of my job offer (it was made to me in a somewhat ambiguous telegram from Licklider), and as I began to rue finding out in this fashion about his fabled disinterest in administrative details, Freeman said that any friend of Licklider's was a friend of his, or words to that effect. He would immediately make my appointment official, and, moreover, because MIT was on a July-1 12-month basis for salaries, I could go to the bursar's office on the morrow and pick up my summer's pay. (Small wonder I blithely followed the man to BBN.)

73. J. A. Swets, "Some Possible Uses of a Small Computer as a Teaching Machine," *Datamation*, vol. 10, no. 6, June 1964, p. 39.

74. W. Feurzeig, P. Munter, J. Swets, and M. Breen, "Computer-Aided Teaching in Medical Diagnosis," *J. Med. Educ.*, vol. 39, no. 8, Aug. 1964, pp. 746–754; J. A. Swets and W. Feurzeig, "Computer-Aided Instruction," *Science*, vol. 150, no. 3696, 29 Oct. 1965, pp. 572–576.

The second reference above illustrates the ability of BBN staff to publish articles of general scientific interest in academic journals. Kryter, Kristofferson, and Senders also published in *Science* during this period.

Apropos, BBNers have historically been available to serve on committees assembled to apply science to national problems. As an example, Kryter, Green, and I were on a committee of the National Research Council to recommend a standard fire-alarm signal. The committee recommended a repetitive temporal pattern — dot, dot, dash . . . dot, dot, dash — that could be used with whatever physical signal would best combat the ambient noise of a particular setting (apartment building, factory, shopping mall, hotel, etc.). The recommendation has been adopted as a standard (simplified to three pulses of equal length), both in the United States and internationally, but is not so far in use. J. A. Swets, D. M. Green, et al., "A Proposed Standard Fire-Alarm Signal," *J. Acoust. Soc. Am.*, vol. 57, no. 3, 1975, pp. 756–757. The past few years I've been on a National Research Council committee that used signal detection theory to evaluate the accuracy and efficacy of the polygraph for lie detection. *The Polygraph and Lie Detection*, National Academies Press, Washington, D.C., 2003.

75. Subsequent support for several extensions and refinements came from the Office of Naval Research, under a six-year contract monitored by Drs. Glenn Bryan and Victor Fields.

76. An ARPA contract through the Air Force Office of Scientific Research ran from 1967 to 1974. The project falls outside of the period set for this chapter, but I'll include it because I know it best. Drs. Charles Hutchinson and Glen Finch were Air Force monitors; Drs. Lee Huff, Austin Kibler and George Lawrence advised the work from ARPA. The final report was prepared by D. N. Kalikow: BBN Report 2841, 1974.

77. J. R. Carbonell and M. M. Klatt, *Toward a Computer-Based System for Teaching and Testing Syntax and Semantics*, in BBN Report 1575, 1967, pp. 93ff.

78. D. H. Klatt and D. Dodds, "Computer-Controlled Display for Second Language Learning," *J. Acoust. Soc. Am.*, vol. 45, no. 1, Jan. 1969, p. 324(A).

79. B. Fraser and M. M. Klatt, *A Partial Inventory of Phonetic Difficulties Encountered in Learning a Second Language*, in BBN Report 1575, 1967; R. J. Carter, *An Approach to a Theory of Phonetic Difficulties in Second-Language Learning*, in BBN Report 1575, 1967; K. N. Stevens and M. M. Klatt, "Analysis of Vowels Produced by Spanish and English Speakers," *J. Acoust. Soc. Am.*, vol. 46, no. 1 (Part 1), July 1969, p. 110(A).

80. D. N. Kalikow and J. A. Swets, "Experiments with Computer-Controlled Displays in Second-Language Learning," *IEEE Trans. Audio Electro-Acoustics*, vol. AU-20, no. 1, Mar. 1972, pp. 23–28.

81. R. S. Nickerson, D. N. Kalikow, and K. N. Stevens, "*Computer-Aided Speech Training for the Deaf,*" J. Speech and Hearing Disorders, vol. 41, 1976, pp. 120–132.

82. M. S. Mayzner and W. R. Goodwin, "Historical Perspectives," *Minicomputers in Sensory and Information-Processing Research*, M. S. Mayzner and T. Dolan, eds., Erlbaum, Hillsdale, N.J., 1978, p. 21. The potential of the system to integrate studies of perception, learning, thinking, and problem solving was pointed up by Lincoln Laboratory psychologist B. W. White, "Studies of Perception," *Computer Applications in the Behavioral Sciences*, H. Borko, ed., Prentice-Hall, Englewood Cliffs, N.J., 1962, pp. 302–303. What was innovative about this effort, I hope I've made clear, was to the credit of Ed Fredkin and Bill Fletcher.

83. J. A. Swets, D. M. Green, and E. F. Winter, "Learning to Identify Nonverbal Sounds," *J. Acoust. Soc. Am.*, vol. 33, no. 6, June 1961, p. 855(A).

Chapter 4

Early Years of Basic Computer and Software Engineering

Compiled by David Walden

Other chapters in this volume cover how computers have been used in particular application areas: psychology, educational technology, medical applications, speech processing, natural language understanding, data networking, distributed systems, control systems, and signal processing and detection systems. This chapter describes BBN's early activities that were more aimed at creating the computing capabilities themselves.

From the time J. C. R. Licklider and his people arrived at BBN, there have been several characteristics that defined BBN's approach to advancing computer and software engineering. BBN has:

- Been a "lead user" (to use von Hippel's term[1]), pushing the state of the art with prototypes, early quasi-production systems, or full production systems, not merely waiting for someone else to bring out the next product.
- Sought interactivity and high performance.
- Been a "technology integration" company, developing (or modifying) hardware or software as appropriate — not a "systems integration" company, which just cables existing stuff together.
- Always been well ahead of the mainstream; for example, ahead in moving beyond mainframes and batch processing, traditional telephony-based data communications, and big-company approaches generally.
- Characteristically connected often into the real (i.e., analog) world.
- Had a desire for growth and impact and thus an inclination to entrepreneurship.

A companion chapter (Chapter 21) covers additional operating system and language work as well as describing BBN's computer-building activities, all of which provide further illustrations of the above points.

4.1 The PDP-1s and PDP-1 time-sharing

Waldrop's book on Licklider[2] tells the well-researched story of Licklider coming to BBN, Ed Fredkin's later arrival, and their joint impact on BBN's early computer systems. In this book, Leo Beranek and John Swets (Chapters 1 and 2) have summarized the story from their points of view. Much of the information in this section comes from Ed Fredkin himself.[3] Thus, this section is substantially from Ed Fredkin's point of view.

Fredkin Joins Licklider at BBN

Ed Fredkin got out of the Air Force in "late 1957 or early 1958" and went on the MIT Lincoln Laboratory payroll. He was already at Lincoln Lab at the time, having ben posted there by the Air Force. Among other things he did at Lincoln Lab, Fredkin ran little courses to teach people about computers. He remembers trying to convince people to convert from octal programming to assembly language programming, not always successfully.

In 1958 Fredkin decided to start his own company, and for his anticipated venture he ordered a computer, a Royal-McBee (formerly Librascope) LGP-30. This computer executed 60 instructions per second and included (fixed point) multiply and divide instructions. It had four vacuum tubes and one board with about 1,500 diodes on it. If you cut a particular diode out of the circuit, the machine kept working perfectly but now had a square-root instruction.

Even though Fredkin had ordered the computer, he didn't have the money to pay for it. He talked to friends like Frank Heart (at Lincoln Laboratory) and Tom Marill (who had been at Lincoln Lab and had gone to BBN). Marill suggested the possibility of Fredkin getting business for his new company from BBN.

So Fredkin went to see Licklider at BBN. Licklider told Fredkin to "Come work at BBN" and said that BBN would accept the Royal-McBee LGP-30 Fredkin had on order. Licklider suggested that Fredkin could teach people at BBN about computers. Fredkin had only been on the Lincoln Lab payroll for three months when he left for BBN.

At BBN, however, executive vice president Sam Labate wanted a reason why BBN should buy a $50,000 computer (perhaps $250,000 in 2002 dollars, according to Fredkin), so the decision was made that Fredkin would make the computer do BBN's payroll function. Once at BBN, Fredkin started to write the payroll program. As part of this effort, he asked various people what they wanted the program to do. What people said they wanted was practically artificial intelligence. Fredkin asked them, "Are you sure you need that?" They said yes. After a while, Ed suggested that IBM be called in to provide the payroll function. IBM brought a tabulator machine using plug boards for programming, and that (totally non-AI) system satisfied BBN's payroll needs.

Fredkin remembers that when he got to BBN there was no one who really knew computers. If you said "computer," someone would ask, "analog or digital?" as a way of showing they at least knew something about computers.

At BBN Fredkin provided insight about computers to Jerry Elkind, Tom Marill, and others. One of the people Fredkin taught about computers was Licklider. However, Lick's programming instincts were not so good. According to Fredkin,

> He tended to focus on the wrong stuff. For instance, he spent time thinking about how to solve the problem which caused others to invent index registers. The PDP-1[4] didn't have any. I tried to explain to Lick that the problem had been solved on other computers (such as the IBM 7090), so Lick didn't really need to re-invent [index registers]; but I wasn't successful.

Fredkin continues,

> There was nothing I could do to get Lick to be a good programmer. He insisted on being a coder, and he had wonderful high-level ideas; but what he always chose to code never made sense to me. I tried to straighten him out a number of times but couldn't succeed. It's just that at that time he didn't have a knack for coding, either in the style of code or in the things he chose to code.

In any case, says Fredkin,

> Lick learned a lot from playing with the LGP-30 and PDP-1.
>
> [His] experiences with the computer enabled him to share the kind of vision that

> John McCarthy brought to BBN. Many of McCarthy's ideas (along with a few of mine) contributed to Lick's vision of Man-Computer Symbiosis.

Fredkin also remembers how much he learned from Licklider.

> Lick taught me a lot about doing science and engineering in general. Working for him was a fantastic educational experience for me. Lick was the first person I had ever encountered who seemed to realize that I had a lot of potential. He educated me, trained me, accepted my advice (as to buying the LGP-30 and the PDP-1 and to having BBN hire people I recommended), and much more!
>
> It is hard to convey how wonderful it was working for Lick. He was a really unique person, absolutely brilliant, and actually daring. Hiring me and supporting me so I could do the things I did took guts and self-confidence.
>
> He had many, many ideas, some quite original. For instance, Lick thought that every computer should have two screens: one flat so you could write on it with a light pen like writing on a desktop, and one vertical for normal viewing.

As mentioned in Chapters 1 and 5, Licklider had the principle that you should not hire anyone who doesn't improve the average intelligence of the group. Fredkin says that at one point he wanted to push further into computers than Licklider did and wanted to hire a computer architect. But Licklider wanted to know if this potential employee was the best computer architect in the world. Fredkin thought the potential employee was really good, but the prospect wasn't old enough yet to have shown he was the best. Licklider asked who the best computer architects were, and Fredkin told him John Cocke and Wes Clark. Licklider said that Fredkin should hire one of them. Fredkin tried to hire Cocke or Clark but didn't succeed. Meanwhile, the Gordon Bell, the potential employee Fredkin wanted to hire, went to Digital.[5]

Fredkin Gets Excited about the PDP-1

The Eastern Joint Computer Conference was in Boston in December 1959, and there Fredkin saw the PDP-1 and "realized it was fantastic."

The project to design and build the PDP-1 had started just four months prior to that. Ben Gurley was the designer, and he also built the computer with the help of one engineering assistant.[6]

"So," says Fredkin, "I convinced BBN to become Digital's first customer for the PDP-1. We arranged to borrow the prototype PDP-1 while Digital built the first 'production' version." The prototype machine (a PDP model 1a) was delivered to BBN in early 1960 (see Chapters 1 and 3 for more about this installation).

At the time, according to Fredkin, Digital had the idea that a computer manufacturer shouldn't do software. Thus, Fredkin wrote the assembler for the machine, wrote utility routines (like a rudimentary operating system), and so forth. In that era programs were typically written on ruled sheets with columns for the various fixed-length instruction fields. Ed's assembler for the PDP-1 was called FRAP, which stood for "free of rules assembly program" and which allowed variable-length instructions. At some point Fredkin also wrote light pen software for the machine.

Eventually the first production model PDP-1, a PDP model 1b system (serial number 2), arrived at BBN. That was Digital's first PDP-1 sale. Fredkin says,

> When the PDP-1b arrived, BBN had a ceremony with a lot of hoopla. I wrote a little program so that the PDP-1b could cut its own ribbon at a ribbon-cutting ceremony (appropriate since one of the ideas was for the PDP-1 to interact with the real world). Digital founder Ken Olsen was present.

Bill Fletcher also was working with Fredkin on the PDP-1b. Fredkin and Fletcher had met when they were seven years old. They had both gone to Cal Tech, and they had both become fighter pilots. Bill was stationed on Long Island, and Fredkin convinced him to quit the Air Force and come to BBN. Fletcher was involved in many BBN projects while he was at the company, including TELCOMP (see page 63).[7]

Fletcher remembers[8] going to work immediately upon his arrival at BBN on I/O "stuff" for the 1b, to make it a multiuser time-sharing system. Fletcher also remembers,[9]

> For a long time the PDP-s didn't even have a divide instruction. The machine came with a "divide step" which was used 17 times in a row and then detailed software had to be written to take care of signs and over/underflows. I wrote the final, fastest and smallest subroutine to perform the fixed point divide which was the only thing available within the floating point subroutines. Shortly after the time that DEC (Ben Gurley) released the hardware fixed point feature for the PDP-1 we discovered that there was a seemingly random, small, infrequent error in the floating point divide that I eventually discovered was caused by a bug in my fixed point divide subroutine (BBN hadn't purchased the hardware divide feature when it was offered). When I distributed the fix I was surprised to find out that Ben had copied my subroutine verbatim to produce the hardware feature so he had to correct the hardware also.

By that time there were a number of PDP-1 machines in the field.

PDP-1b Time-Sharing: the Research Computer System

Fredkin says that he had the idea of hiring John McCarthy and Marvin Minsky. This soon led to plans to develop a time-sharing system for the PDP-1. McCarthy had the idea of time-sharing, and Fredkin saw the potential for time-sharing on the PDP-1. McCarthy says,[10]

> Around 1960 I began to consult at BBN on artificial intelligence and explained my ideas about time-sharing to Ed Fredkin and J. C. R. Licklider. Fredkin, to my surprise, proposed that time-sharing was feasible on the PDP-1 computer.

Fredkin says,

> John's invention of time-sharing and his telling me about his ideas all occurred before the PDP-1 existed. When I first saw the PDP-1 at the Eastern Joint Computer Conference, I realized that it was the perfect low-cost vehicle for implementing John's ideas. That is why I specified that several of the modifications for time sharing be part of the PDP-1b.

Fredkin knew Ben Gurley (from Lincoln Laboratory), who was the machine designer at Digital; thus, Fredkin also suggested or designed improvements for the hardware with time-sharing in mind. For instance, Fredkin designed the input/output system to be the first to have a modern interrupt system (what Digital called "sequence break").

> Up until then, computer designers had the idea that asynchronous events should not interrupt the program just anywhere. The TX-2 interrupt systems had a bit on every instruction that could be set to say whether an asynchronous event could interrupt that instruction. An IBM technique was something called a "trap transfer," which when executed accepted an interrupt at that point if one was pending. People didn't think of interrupts that could interrupt the state of the machine at any time.

Fredkin's interrupt system saved the state of the machine (a four-register block[11]). Fredkin programmed all input/output routines to be completely asynchronous except the CRT. Fredkin also suggested or designed other instructions, designed the character set, selected fan-fold paper tape (one of two options being considered), and he designed nonspacing characters for the Soroban typewriter (so it was possible to type characters such as a less-than-or-equal-to sign). Fredkin also designed a real-time analog input/output system with A-D and D-A converters and with 18 computer-controlled relays.

McCarthy states,[10]

> Fredkin designed the architecture of an interrupt system and designed a control system for the drum to permit it to be used in a very efficient swapping mode. He convinced Ben Gurley, the chief engineer for D.E.C., to build this equipment.

According to Fredkin, when McCarthy explained his ideas for time-sharing, he said that time-sharing should be done in RAM with interrupts as is done today. However, at the time no one could afford a big enough RAM: it was $1 per bit then, which would be $6 or $7 per bit now.[12] Fredkin had an idea about how to demonstrate that McCarthy's idea was right: Fredkin invented the swapping drum. The basic drum came from Vermont Research. The PDP-1 initially had 4,096 18-bit words of memory. In Fredkin's design, the swapping drum could read in 4,096 words while simultaneously writing out another set of 4,096 words in 20 milliseconds (5 microsecond cycles). He studied the timing diagrams of the computer RAM read-write cycle and the drum read-and-write timing and figured out how each read-write cycle of the computer could read one word from memory to the drum and then write one word from the drum to memory. Furthermore, because the drum was 4,096 words around and memory had 4,096 words, it was possible for the system to notice where in the 4,096-word interval the drum was relative to its read-write positions and to start the transfer at that same point in the 4,096 words of memory; in other words, there was no latency waiting for the drum to spin to the beginning of a 4,096-word drum block. However, there was a problem. To be synchronized with the machine and have 4,096 words around it, the drum need to rotate at 3,000 rpm. Unfortunately, Vermont Research couldn't find a motor of the right design and right speed, so the actual system ran a little slower than Fredkin's optimized design. In Fredkin's view, it is unfortunate that, rather than seeing his use of such an optimized swapping drum as an indication of what a time-sharing system could do if it was all in RAM, the whole world copied the idea of a swapping drum but without having a drum that worked like Fredkin's. Thus, later time-sharing systems had lots of latency and were very slow.

Fredkin continues,

> The hardware suggestions were mostly in the PDP-1 before it arrived. However, the swapping drum was added later. It took quite an effort to convince Digital to do it. There is a great story about that event. The second PDP-1 went to MIT, where Professor Jack Dennis led a group of students who implemented a lot of good software. He also wanted a "swapping drum" to do time-sharing. I kept pestering Gurley to offer to build it, but he never got back with a proposal. One day, McCarthy, Gurley and I were all at MIT and John and I suddenly started pestering Gurley to agree to build the drum system I designed. Gurley's response was that we hadn't ordered it. John and I both said something like "You mean, if we order the swapping drums right now, then you'll build them?" Gurley laughed and said "Yes." John said "Wait right here." He ran down the hall to Professor Zimmerman's office (I think) and got them to give him a PO number from RLE at MIT while I got on the phone to Licklider and asked him to get me a BBN purchase order number. Amazingly, in about 15 minutes, Gurley had two PO numbers and agreed to build a swapping drum system for both MIT and BBN, which he did.

Of his departure from BBN, Fredkin says,

> I left BBN in late 1961. It is easy for me to be certain about the date as it was not long after Jânio Quadros resigned as president of Brazil (August 1961). Rollo

> [Silver] and I had planned to travel to Brazil. We gave notice to BBN that we were leaving. The Quadros resignation caused us to change our minds about going to Brazil. I told Licklider that I didn't have a compelling reason to leave right then, but Lick suggested that since we had put our departure into motion, there wasn't a good reason to change the date. Rollo and I arranged to continue working on PDP-1 software by both consulting to Digital after we left BBN, in the fall of 1961.[13]

Regarding his leaving the PDP-1 time-sharing work, Fredkin says,

> While I worked out the details of the hardware and software designs, I left before much of the time-sharing-specific software had been implemented. I didn't leave much documentation, so my impression was that McCarthy directed Boilen to work on the implementation.

McCarthy says,[14]

> It was planned to ask NIH for support, because of potential medical applications of time-sharing computers, but before the proposal could even be written, Fredkin left BBN. I took technical charge of the project as a one-day-a-week consultant, and Sheldon Boilen was hired to do the programming. I redesigned the memory extension system proposed by D.E.C. and persuaded them to build the modified system instead of the two systems they were offering, but fortunately hadn't built. I also supervised Boilen. . . .
>
> My recollection is that the BBN project was finished first in the summer of 1962, but perhaps Corbato remembers earlier demonstrations of CTSS. . . . BBN didn't operate the first system and didn't even fix the bugs. They had few computer users and were content to continue the system whereby users signed up for the whole computer.

Bill Mann says,[15]

> I started at BBN in June of 1962, after my freshman year at MIT. John McCarthy (then with the MIT AI group) got me the job, originally for the summer.
>
> Fredkin left about when I started; the time-sharing system had only been designed. McCarthy only consulted, and contributed little to the implementation.
>
> The project was the first time-sharing system, developed on PDP-1 serial number 2, with 12K (?) of 18-bit memory and a Vermont Research swapping drum. This was installed downstairs at 50 Moulton Street.[16] The previous machine at BBN was an LGP-30, which was still around but not being used (much?) when I arrived.
>
> The [principal investigator] was Licklider, but we rarely saw him. Shelly Boilen was the project lead. He, Lick, and possibly McCarthy had designed the project, but little or no coding had been done. I think we used the Macro assembler from MIT. I was a green freshman who had just spent five months at MIT totally immersed in PDP-1 and TX-0 programming. I was living in Belmont with Alan Kotok of DEC and two other roommates, and commuting to BBN by bus or motor scooter. I spent my spare time at MIT, at the Tech Model Railroad Club.[17]
>
> By the end of 1962, Shelly and I had gotten the time-sharing system working for five users, although there were rarely or never five people who wanted to use it at the same time. Each user got 4K[18] plus some operating system services. The terminals were Selectrics.
>
> I probably did half the coding, but none of the design. The system was demoed in September 1962. The time-sharing system was stable, but the demand on the machine was light and I don't think it was used much. I think we had 5 Sorobans by the time the project was wrapped up.
>
> Lick et al. (not including me) published a paper.[19] The paper says, "The purpose of the BBN time-sharing system is to increase the effectiveness of the PDP-1 computer for those applications involving man-machine interaction . . . ," hence (probably) the

> word "debugging" in the title. The system was built to support non–computationally intensive tasks such as "debugging, small calculations, text editing, teaching programs" — in other words the kind of programs that could share a computer without interfering with each other's time to completion. The paper includes a long section on DDT (called TYC), which was unpublished at that time.

In the paper, McCarthy also says,[20]

> [the system] has been in operation at BBN since September 1962. . . . is operated four hours per day . . . five [typewriters] . . . weaknesses: There is no program library on the drum (nor mag tape, only a paper tape reader). . . . Versions of the utility programs especially adapted to time-sharing are desired.

Jon Cole (who arrived at BBN in September 1962) remembers Bill Fletcher and Jack Brown installing a plywood false floor and running wires and other gear for the PDP-1b, perhaps when it was moved from its first location downstairs at 50 Moulton Street to its second location in the same building. Also according to Cole, Model 28 TTYs with 5-bit Baudot code terminals later replaced the Sorobans, and still later Model 33 TTYs with 8-bit ASCII were used.

Mann continues,

> The most exciting story was the time, near the end of the project, when someone decided that we needed a computer center manager and appointed Louis (Lew) Clapp, who was a noncomputer scientist from an acoustics group. DEC had had a lot of trouble with that PDP-1 (it was really a prototype) and frequently engineers or techs made wiring changes, which they carefully noted on a huge set of prints [of the PDP-1 wiring]. One Friday evening they went home, leaving the prints spread out on the floor at the back of the computer room; Lew came in, saw them, and threw them out, on the theory that people who made a mess in his computer room should be punished. He was immediately fired. A few weeks latter he was rehired by the acoustics people.

McCarthy's assessment that BBN didn't even fix the bugs in the time-sharing system and continued to run the PDP-1b on a stand-alone basis has some truth to it but also is misleading. In the years that followed, the PDP-1b (known throughout BBN as the Research Computer System) was extensively used. The hardware that supported time-sharing was also the basis for the time-shared PDP-1 TELCOMP system (page 63) and the time-shared PDP-1 LISP system (see Chapter 21); in both cases, the language system took up the whole machine and supported time-sharing among its users through time-sharing mechanisms integrated into the language system. At other times, some users did use the machine on a stand-alone basis, depending on the needs of the application. According to Jon Cole, this machine was used for lots of experimental psych work.[21] These three uses (TELCOMP, LISP, and stand-alone) competed furiously for machine time.

Having an interactive PDP-1 at BBN also attracted some key researchers, among them Wally Feurzeig (see Chapter 13), who was also directed to BBN by McCarthy.

PDP-1d Time-Sharing: Hospital System

Being able to talk about time-sharing at BBN led to BBN's next time-sharing system.[22] Jordan Baruch was doing acoustics work at the Clinical Center at NIH, first involving vibration control and later involving instrumentation for the cardiac and neurological wings. Thus, he was traveling to NIH weekly. At night he would go over to the home of Jack Masur, director of the NIH Clinical Center, where Masur would provide Baruch with gin and jelly beans. One night Masur told Baruch he was to give a speech the next week to some women about the place of computers in health care and asked him what he thought about it. Baruch described his vision about the possibilities for patient medical records and various other interconnections, while Masur fed him more gin. The next week, after the speech, Masur phoned Baruch and told him the speech had been a hit — everyone was enthusiastic. Then he said to Baruch, “You should go do it.” Baruch said that building this kind of system would be expensive. Masur told him to apply for a grant in the Division of General Medicine. BBN got the grant — $1 million over three years, Baruch remembers. Baruch wasn’t planning to run this project — he thought Licklider would do it. However, Licklider didn’t want to, so Baruch was the one to run it.

The Hospital Project started, says Jon Cole, with a proof of concept done on the PDP-1b machine, consisting mainly of showing that ASCII TTYs with long-distance copper wires could work with the time-sharing system. When the Hospital Project got funding past the proof of concept, it ordered a PDP-1d. It was DEC serial number 45.

Initially, Exec II (BBN’s second time-sharing system) was written for the 1d, led by Nancy Haggerty, according to Jon Cole. However, Exec II never worked reliably.[23]

Thus (still according to Cole), Steve Weiss, Andy Munster, and others, planned to build a new, much more cleanly organized time-sharing system, known as Exec III, with state table for users, etc. The Exec III time-sharing system was extensively documented,[24] and the PDP-1d and Exec III ran for a *long* time after the end of the hospital application project, residing in the back room of BBN’s 20 Moulton Street building. The system had a complete suite of development tools, which were relatively easily configurable for different applications. Much Logo work was also done on this machine (see Chapter 13). Perhaps the PDP-1d’s most famous use was as the software development machine and operational data collection machine for BBN’s ARPANET project (see Chapter 17).

Bill Mann remembers moving to the the Hospital Project.

> I decided not to return to school full-time, and Shelly and I joined Jordan Baruch for the Mass General Hospital Project. This needed a more reliable machine, so I worked with DEC to add a few extra instructions (lch, dch, sni, etc.) to a new PDP-1,[25] which had 24K of memory, another Vermont Research drum, a huge, customized FASTRAND drum and tape units, and a multi-line teletype interface box.[26] This was installed in a newly acquired neighboring warehouse [20 Moulton Street], where it lived for many years. Later John McCarthy, then at Stanford, ordered an identical machine from DEC . . . one of the last few PDP-1’s made. . . . I configured MIDAS (changed the sequence break channel assignments) for that machine one Saturday afternoon at [Digital’s] old mill [in Maynard], fixing a couple of hardware bugs while I was at it (I added two terminating resistors, leaving a note for the DEC engineers, who had left for the day). . . .
>
> The Hospital Project staff included Paul Castleman (his father was an MGH doctor), Steve Weiss (a brillant MIT undergrad who later went to Michigan), John Hughes (an independently wealthy manager, who owned an ocean-going catamaran and took long leaves to sail it), and a half-dozen others . . .
>
> Shelly was the lead technical guy; he was smart and had good ideas, but even though he had been an English major (maybe at Antioch), he had a terrible time communicating. One of my jobs was explaining what he meant to the rest of the group. Jordan was a joy to work for, he kept giving me raises every three months, was smart, friendly, and told wonderful stories; Tom Marill (somewhat snidely) called him the world’s greatest salesman. One weakness was that [Jordan] kept hiring . . . people who were generally smart but knew little or nothing about programming. So I ended up doing a lot of coaching.

Another member of the project was Bob Morgan. Steve Weiss and Bob Morgan both lived at Senior House at MIT. Weiss recruited Morgan to join BBN in spring 1964 as a part-time employee while he was still a full-time MIT student. Morgan says,

> I was working with Steve Weiss, Dave Walton, and to some extent Bill Mann. I was working on the time-sharing component of the Hospital Computer Project — in particular the I/O processor.[27]
>
> Bill Mann was doing the JobHunter project which was an interactive question-answering system. Steve Weiss was doing the scheduler and utility libraries. I can't remember what Dave Walton was doing. I was doing the I/O processor for the timesharing system. Andy [Munster] was there. Paul Castleman and Sally Teitelbaum were working on applications as was Nancy Hurley [and others].

Bill Mann's view is,

> There was no hope that the project would be anything but a prototype [as a hospital application] — the hardware was not in any way adequate. The software was a first cut. The feelings at MGH ranged from "very interesting" to "it may kill my patients, get it out of here," with a strong bias toward the latter. I worked on system software, but not directly on the time-sharing system. Macro had been replaced by MIDAS (originally written by Bob Saunders at MIT); I had taken over maintenance and extensions, including DDT and a relocating linking loader.[28,29] I also coded the Medication Order function, and helped with general integration and debugging. I got a lot of calls at home nights.

When Bernie Cosell got to BBN in late 1965, the Hospital Project was already running under Exec III. Steve Weiss drafted Bernie to help finish the system, and Bernie got left maintaining it when Weiss and Bob Morgan left BBN to go to grad school. Also working on the project at that time, according to Cosell, were Andy Munster and Jon Cole.

PDP-1c(s)

Yet another model of PDP-1 was used at BBN. These machines came after the PDP-1d, according to Jon Cole. When the TELCOMP service (see page 63) decided to buy one or more dedicated PDP-1s, they bought PDP-1c machines from DEC, which went into the New Jersey office, Los Angeles office, and in London, says Bill Fletcher. These didn't have character instructions, didn't have the capability for the UNIVAC I/O units, and so forth. In time, TELCOMP decided to switch to PDP-9s. However, the first of these systems were PDP-7s because they were available faster and were easy to convert the TELCOMP code to.

4.2 Higher-level language work

LGP-30 compiler extensions

The earliest reference to high-level language work at BBN is by Richard McQuillin.[30] This is part three of a four-part study for the U.S. Bureau of Ships. The other three parts had to do with vibration-damping techniques.[31] McQuillin's report describes what is apparently an addition to the ACT 1 compiler that came with the LGP-30,[32] to permit arithmetic with complex (fixed or floating point) numbers.

DECAL

DECAL was a combination of an assembler and compiler.[33] Apparently you could intermix lines of assembly code with lines of compiler code; for example,

```
lac a
dac c[i,j]
if c[i,j] > k then goto p
```

Ed Fredkin states that he designed DECAL; however, Digital gave the job of implementing DECAL to Dick Bennett of Data Processing Inc. in Waltham, Massachusetts. Bennett did the work in 1960.

Bennett was an early believer that people should make money off of software. Thus, when he delivered the finished DECAL product to Digital, he delivered only the binary object code and "a very condensed manual," but no symbolic source code listing. Digital had been expecting source and object code, and there was an argument between Digital and Bennett. Someone (maybe Fredkin) suggested that as a way out of the argument, Digital could to pay Bennett for the product and receive only the object code, but Digital would also then have unrestricted rights to do whatever it wanted with the program. The deal was made, and Bennett was paid.

Then, according to Fredkin, Digital brought the program to Fredkin at BBN. Roland Silver had written a trace program, and Dick McQuillin got involved and had to finish the job when Fredkin left BBN. DECAL was disassembled and reconstructed and the program with symbolic source code was delivered to Digital.[34] However, Buzz Bloom remembers that "we discarded entirely what was done by Dick Bennett and started over from scratch."

According to the Preface to the DECAL Programming Manual (by Richard McQillin), BBN's job was "to provide symbolic listings and a more elaborate manual, and also to implement certain improvements to the system." BBN started work in May 1961 with Fredkin supervising Buzz Bloom and David Park. In November 1961 Fredkin left BBN, and "the project terminated, having achieved the implementation of a number of new features in the Compiler and the Linking Loader. Further work was subsequently done by Fredkin, yielding the binary paper tape known as F17C; and progress toward a programming manual was made."

DECAL used paper tape for input and output. Bloom says, "It was a one-and-a-half-pass compiler that permitted forward referencing of symbols. The trick was to load the output tape from pass one backwards (in reverse order) so that the compiler defined locations associated with symbols occurring after references would be read before the compiled code of the reference."

It is not clear to me whether this linking-loader trick was first used in DECAL or had already been used with an assembler for the PDP-1. The problem was that the PDP-1 was originally planned to have a low-priced version with only 1,024 words of memory. Ed Fredkin reports that he constructed a system that worked as follows. As the assembler read instructions in, it put symbols in the next available spot in the symbol table. Once in the symbol table, symbols were never moved. A search was done through the existing symbols in the table: if the symbol was already in the table, the table end pointer was not moved, effectively discarding the redundant version of the symbol; if the symbol was not already in the table, the table end point was advanced to include the new symbol. The system used no auxiliary storage and punched a paper tape as it assembled the program. Forward references to symbols in the program were handled by outputting the assembled instruction along with the symbolic version of the symbol onto the paper tape; when the symbol definition was later read in, the symbol was added to the symbol table and the symbol with its value was then punched out on the tape. When the assembler was done reading in the symbolic program and punching out the paper tape of the binary, the fanfold paper tape was flipped over and read into the linking loader in the reverse direction from which it had been punched (something uniquely possible with fanfold which, Digital computers used instead of rolled paper tape during the paper tape era). As the paper tape was read by the loader, locations of undefined symbols from the assembler pass were read before the instructions that referenced them and the assembly of the instruction could be completed. "It was such fun then," says Fredkin.

In June 1962 work began on BBN-DECAL, the system described in the DECAL manual by Richard McQuillin[35] and funded by BBN, the Council on Library Resources,[36] AF/CRL, and eventually Digital. This work was done by McQuillin, Bill Fletcher, David Park, and Craig Fletcher, with assistance from Harrison Morse of Digital. In late September 1963, BBN delivered to Digital and DECUS (Digital's user group) the completed system, a complete set of manuals,[37] a complete set of listings, and so on. The title page of McQuillin's manual says, "Submitted to Digital Equipment Corporation...Attention: Mr. Gordon Bell."

When I asked Bill Fletcher about his involvement in DECAL, he told me that I already had the story pretty much correct. He did note that if I looked in the back of the DECAL manual at the list of error codes, I would find two error codes that Bill Fletcher and his brother Craig put in as a way of including their initials in the manual: `cmf` for compiler malfunction and `wef` when the paper tape was put into the reader backwards. `cmf` is on page 65; perhaps `wef` is in the linking loader manual.

JOSS to Logo

A series of programming languages was implemented at BBN in the 1960s,[38] starting with a PDP-1 version of JOSS,[39] through a succession of interrelated languages, illustrated in Figure 4.1. This subsection discusses the creation of these languages.

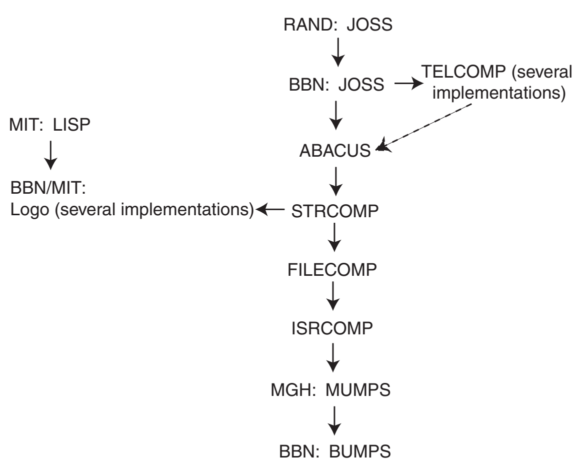


Figure 4.1 Some of BBN's language projects.

Bob Morgan was introduced in the earlier section on the PDP-1 time-sharing system. Although he left full-time BBN employment for graduate school, he continued working summers implementing ABACUS and the first machine language interpreter for Logo. After getting his PhD in math from MIT, Bob returned to BBN from 1973 to 1981 and since then has worked for several other companies, always concentrating on compilers, particularly optimizing compilers.[40]

Bob was present at the beginning of BBN's work with this series of languages, and I asked him to recount what he remembered. As Bob related the story, he emphasized that he personally shouldn't be the focus: "Steve Weiss . . . [was] the primary [person] on the early JOSS work. . . . I am simply the chronicler":

> Cliff Shaw was coming on a site visit for the Hospital Computer Project. On a steak dinner bet, Steve Weiss, Dave Walton and I decided to implement a JOSS interpreter for the Hospital Project to impress [Cliff] Shaw. We had three weeks to do it before Shaw's visit (and it clearly ended up being a subset). Steve was the real lead on this. It was many nights without sleep but we pulled off enough to do the "snow job" that we had intended. I don't think we ever got the steak dinner from Jordan Baruch [leader of the Hospital Computer Project].
>
> The JOSS-on-a-bet system was done during the summer of 1964 (I think July and August). The PDP-1[d] time-sharing system was organized as 4k (18bit) words for the time-sharing system, 4K (18bit) words for utility libraries, and several 4K (18bit) word [blocks] for multiple program executions (and there may have been another 4K for I/O, but I don't remember exactly). One of the utility routines was a rudimentary syntax analyzer which was used to get some of the parsing done. Another set of utilities was a floating point package which was used for the computation. The system was implemented to reparse and execute each statement as it was executed: no intermediate code was saved because there was no room.
>
> It was a sufficient success that, when Bill Fletcher saw the system, he realized that he could make a real system out of it. This became the TELCOMP product a year or so later. He did a complete reimplementation [for the PDP-1b] with a dedicated time-sharing system tightly interconnected with the execution of each TELCOMP session — in other words, as a dedicated time-sharing system with no separation between the interpreter and the time-sharing system.
>
> I took the TELCOMP system that Bill Fletcher et al. built, removed the time-sharing system, reused the utility routines on the Hospital PDP-1 (but did not use the syntax analyzer [because the 1d time-sharing system already provided one]), and brought up a modified form of Bill's TELCOMP, reengineered for the PDP-1d/45 and called ABACUS. Then people started to get ideas about adding all sorts of functionality to the ABACUS system, [functionality that] could be done without disturbing the TELCOMP product. One idea was adding strings: ABACUS plus strings became STRCOMP.
>
> The Massachusetts General Hospital people (users of the Hospital Project) liked STRCOMP so much that after the project was over they developed the MUMPS language[41] based on a stripped-down version of the ideas of STRCOMP. MUMPS has a long and venerable history itself.[42]

Wally Feurzeig lays claim to the idea of adding text strings to ABACUS, although he doesn't remember the ABACUS step, after TELCOMP, on the way to STRCOMP (see Chapter 13). The STRCOMP manual says,[43] "The string-manipulation features in STRCOMP were originally designed by D. Bobrow, W. Feurzeig, and S. Papert; the design was modified by J. Barnaby and P. Wexelblat and implmented by J. Barnaby with assistance from P. La Follette and P. Wexelblat."[44] STRCOMP could also generate new programs (as text strings) that it could then execute.

STRCOMP was extensively used for years within BBN; it was what we used instead of BASIC, and for a lot more. For instance, lots of processing of the log data coming to the ARPANET Network Monitoring Center was done in STRCOMP. In the 1980s, when the PDP-1d was shut down, Randy Rettberg reimplemented STRCOMP under TENEX, to enable the NMC programs to run under TENEX.[45]

As Feurzeig describes in Chapter 13, STRCOMP's successful use in educational environments led, in 1966, to the creation of Logo, which he developed with Seymour Papert and Danny Bobrow.[46] Feurzeig continues,

> A little later it became clear that files were needed, both for educational applications, and for applications to clinical medical operations in the Hospital Computer Project. That resulted in the FileComp language which turned into MUMPS.[47]

Feurzeig describes Logo and its philosophy, purpose, and implementation history in some detail in Chapter 13. Essentially, Logo is a dialect of LISP. As a programming language, it has the following distinguishing characteristics, according to Feurzeig:

> The key organizing idea in Logo is the procedure. Logo procedures are self-contained entities. Procedures can be run as single entities (if they are supplied with the set of inputs they require) and they also can call each other to form complex procedure structures, but, in typical programs written in a reasonable style, the individual procedure components are short and sweet and semantically clear. This in contrast with educational languages like BASIC, where a program is a single extended structure like a long piece of spaghetti.
>
> There is virtually no typing [declaring the types of variables] in Logo. Typing can be valuable for helping to ensure correctness and for making more efficient compilation, but it can frustrate beginning students. We didn't want users to get hung up on type declarations standing in the way of expressing their ideas.
>
> During the era when Logo was born, people spoke of non-numerical programming as a special kind of world. Standard languages were all numerical. We introduced strings in Logo because we wanted to include a richer set of objects, like words, to enable applications in domains involving, for example, English language constructs and operations.
>
> Finally, like LISP, one can write a Logo program that itself can write new Logo procedures and then execute [them].

Independently of the Logo effort, STRCOMP was extended with a file-handling capability and called FILECOMP (see Figure 4.1). This, in turn, evolved into ISRCOMP.[43] ISRCOMP lacked a few of the more specialized facilities of STRCOMP but included the capability to access data files built in the Hospital Project Information Storage and Retrieval (ISR) System.

Commercial TELCOMP

TELCOMP led to an important attempt by BBN to provide a widespread commercial service (see Chapter 6). Bill Fletcher was the key technical person involved with TELCOMP. He has provided some details about the TELCOMP implementation:

> Jack Brown and I worked together to take the original PDP-1b and configure and program it to provide the TELCOMP time-sharing service. The first day of revenue service was September 29th, 1965. Jack was mainly the hardware guy and I was mainly the software guy.
>
> TELCOMP was pretty much an exact copy of JOSS with the only improvement that I recall being that restrictions of symbol/name length and nesting depth [were] . . . only limited by the amount of memory available. Tables, lists and stacks didn't have any predetermined maximum size. Licklider particularly liked to use long names.
>
> The PDP-1 was running time-sharing long before the TELCOMP effort started, and I was very much involved in writing many of the detailed IO routines to support the multiuser time-sharing. TELCOMP came up on the PDP-1 under the previous time-sharing environment. Later, the PDP-1 was modified to support more users when it was used as the vehicle for the commercial introduction of TELCOMP. . . .

> As the commercial time-sharing venture succeeded the PDP-9 was selected as the platform for further rollout. PDP-7s were initially purchased for the first couple because of the lead time for delivery of PDP-9s. Jack Brown and I were both involved in both 2 and 3 with Norm Doelling joining as the senior BBN manager as the project expanded from the PDP-1. Porting the TELCOMP software to the PDP-9/7 was undertaken by a new software group under Paul McIsaac. At that time I was serving as the Technical Director with both the TELCOMP Project software and hardware groups as my responsibility. Jack Brown had gone off to something else not long after Norm Doelling joined the effort. TELCOMP was a very awkward type of project to be run under the research environment at BBN and it was frustrations due to that awkwardness that led me to decide to leave BBN for more real world endeavors. Even so, BBN was a wonderful place to work in those days and I have always remembered it with great pleasure.

Norm Doelling came to BBN to do acoustics. When the PDP-1 came to BBN, Ed Fredkin gave a course on computers and programming to all the staff. At that time Norm was involved in patents and licensing and what to do with all of BBN's novel technology. He asked BBN's board to send him to the Harvard Business School one-term middle management course, but it took him a year or two to make that happen. (The term after Norm went to Harvard Business School, the BBN Board sent Leo Beranek to the HBS one-term advanced management course.)

Norm attended the HBS one-term course in the fall of 1964. When he got back to BBN, Sam Labate asked him to go run TELCOMP, where Bill Fletcher was the technical leader. Norm led the marketing and believes they created the first publicly available time sharing service. (GE also claims to have had the first time-sharing service, but Norm believes that at the time, theirs sold only to other GE units.)

In those early days, the TELCOMP service made four PDP-1 TSS terminals on BBN research computer available via dial-up connection (acoustically coupled, presumably) from 9 a.m. to noon or 8 a.m. to noon, and people loved it.[48] However, there were lots of battles about time. TELCOMP wanted more time to offer for outside sale. The researchers, of whom Jerry Elkind was the main warrior, didn't want to give up the time. Norm says that Elkind's prediction for the future of TELCOMP was accurate, including eventual overcapitalization and then collapse. In time, Norm thinks he remembers, the service was somehow expanded to eight terminals.

Eventually there was the question of what to do next. First they got another PDP-1 for the New Jersey office. Then they considered whether to get the last PDP-7 or the first PDP-9, and they chose the last PDP-7.

Meanwhile, Norm was tiring of the internal battles. He went to Sam Labate and said he wanted to move the activity to Route 128 because the environment within BBN was poisonous for the salesmen (e.g., Tom Welch), who were completely unappreciated by the BBN technical people. Norm talked to Sam. Sam talked to Dick Bolt. Dick talked to Norm. Of the question of going to Route 128, Dick said, "You've been here seven or eight years, but you don't understand the company. You can't have a technology-based activity that doesn't interact intimately with BBN's research base." Tired of the battles, Norm left BBN.

Norm concluded, "You have to understand: TELCOMP was developed by a heavy-drinking crew — Bill Fletcher, Paul McIsaac, me. Much of the design was done in Fantasia's bar."[49]

Dan Murphy has added a comment about the PDP-10 version of TELCOMP, of which John Barnaby was the primary implementor:

> I did a chunk of implementation of the PDP-10 version. When we got TENEX running, we rewrote TELCOMP mostly from scratch to be a normal user program rather than a self-contained multiuser system, as it was in its original implementation.

Acknowledgments

In addition to everyone who is noted in the text or notes, Ray Nickerson and John Swets provided review and additional input. I regret that space limitations prevent mentioning by name many other BBN people who also worked on the projects mentioned in this chapter.

Notes and References

1. Eric von Hippel, "Lead User Analyses for the Development of New Industrial Products," *Management Science*, 34:569–582, May 1988.

2. M. Mitchell Waldrop, *The Dream Machine: J. C. R. Licklider and the Revolution That Made Computing Personal*, Viking, New York, 2001.

3. Phone call of September 12, 2002; e-mails of review, correction, and augmentation of October 1 and October 24, 2002.

4. How BBN got a PDP-1 is described in the next subsection.

5. In this chapter, Digital Equipment Corporation, the maker of the PDP machines, is referred to as Digital, DEC, or D.E.C., depending on how the person providing the information referred to it.

6. For more about development of the PDP-1 and images of various original documents, see `http://research.microsoft.com/users/GBell/Digital/timeline/pdp-1story.htm`

7. In an e-mail of October 21, 2002, Fletcher says, "I came to BBN in August 1960 and left to go to work with Dick Morley at Bedford Associates in January 1968. I spent two or so years in the Los Angeles office around the 1964 time frame."

8. E-mail of February 23, 2003.

9. E-mail of October 10, 2002.

10. John McCarthy, "Reminiscences on the History of Time Sharing," 1983. Available at `www-formal.stanford.edu/jmc/history/timesharing/timesharing.html`

11. Program counter, memory result, accumulator, and input/output, as Bill Mann remembers in an e-mail of July 18, 2003.

12. Today it takes something like eight digits to the right of the decimal point to be able to see the price per bit as anything other than zero.

13. Shortly later, Fredkin started Information International Incorporated (III). Rollo Silver went to MITRE, not III. Ben Gurley joined Fredkin at III and was doing high-potential work there when he was murdered. In a phone conversation of October 24, 2002, Silver said, "Ben was one of the four people who started Digital (the others being Ken Olson, Harlan Anderson, and Dick Best). Gurley was a brilliant circuit designer and conceived the idea of the minicomputer (as is well known, Digital originally was developing and selling modules). Gurley was friendly with a paranoid person at Lincoln Lab who everyone else avoided; unfortunately, that person came to Gurley's house one evening when he was there with his wife and six or seven children and shot him to death." According to the history of large-scale computing on the Livermore Lawrence Laboratory website, Fredkin and Edmund Berkeley hired a private detective to find the murderer, who was caught and declared mentally deranged.

14. John McCarthy. Reminiscences on the history of time sharing, 1983. Available at `www-formal.stanford.edu/jmc/history/timesharing/timesharing.html`.

15. E-mail of February 23, 2003.

16. BBN's main building, designed by Dick Bolt — see photo on page 8.

17. Editor's note: For more on Kotok and the Tech Model Railroad Club (TMRC) and their place in PDP-1 history, see Steven Levy's book *Hackers: Heros of the Computer Revolution* (Anchor Books, New York, 1984). The author of this book, Steven Levy, is a technology writer from California, not the longtime BBN executive, Stephen Levy, who wrote Chapter 6 of this book.

18. Under time-sharing, users programs were swapped in and out of the 4K user space.

19. John McCarthy, Sheldon Boilen, Edward Fredkin, and J. C. R Licklider, "A Timesharing Debugging System for a Small Computer," in *AFIPS Spring Joint Computer Conference*, volume 23, pages 51–57, 1963.

20. Ibid.

21. Described in Chapter 3.

22. The information in this section came from Jordan Baruch (phone conversation of October 28, 2002); Jon Cole (phone conversation for which the date has been lost); Bernie Cosell (e-mails of October 21, 2002); Bill Mann (e-mails of February 23 and July 3, 2002, and July 15, 2003); Bob Morgan (e-mail of April 3, 2003); and Paul Wexelblat (many BBN documents). See also Paul Castleman's companion chapter (Chapter 12) for more about the Hospital application.

23. In his reminiscences, McCarthy says, "[BBN] did undertake a much larger follow-on project [after the PDP-1b time-sharing system] involving a time-shared PDP-1 that was installed in Massachusetts General Hospital, where it was not a success. The computer was inadequate, there were hardware and software bugs, and there was a lack of application programs, but mainly the project was premature." The Hospital Project may not have been a complete success; however, the time-sharing system was a great success, as will be described below.

24. Alexander McKenzie, The Hospital Computer Project Time-Sharing Executive System, BBN Report 1673, April 1, 1968.

25. Bill Mann says, "I worked with DEC to specify the new instructions, but Shelly most likely contributed to the I/O configuration. There was another BBN EE (whose name escapes me) who was also involved in getting the hardware specified and working."

26. Paul Wexelblat says that 1d had a scanner for 64 TTYs and shared code in "memory 16." According to Bill Mann, "All the PDP-1s had interrupts, but there were two options — single level and multi-level. There was no real support for floating point; even fixed point multiply and divide were options. The 1d added a carry bit to facilitate multi-precision and there was a new twos-complement add-with-carry instruction (normal add/sub was ones-complement). Shared code was a 1d feature which allowed a user program (which had to run in 4K) to call shared library routines which were permanently loaded in a separate 4K." Bernie Cosell says the 1d also had a "special operate" group which he understood was designed by Bill Mann. Wexelblat says that he understood the 1d to have a fancier memory system — four ports instead of the usual one port. One port was the usual one, the second port was available for the swapping drum, the third port was used by the I/O system (the two tape drives and the Fastrand), and the TELCOMP folks also used the fourth port. Wexelblat also says that the PDP-1 originally had MUS (multiply step) and DIS (divide step) instructions, but DEC converted these to full Multiply and Divide instructions for the 1d.

27. Morgan became full-time at BBN in 1965, and then returned to MIT full time for graduate school in 1966.

28. In time, BBN made use of all three of the software tools that grew up in the TX-0, PDP-1, and TMRC community in Cambridge: the MIDAS assembler with its powerful macro capability, the so-called DEC Debugging Tape (DDT) program, originally written by Alan Kotok, for doing symbolic debugging of programs in memory, and Dan Murphy's TECO (Tape Editor and COrrector) program (written at MIT before he came to BBN) for program editing. While all of these originally worked with paper tapes, each was adapted to serial ASCII files on magnetic tape or on hard drives. Sandy Libman implemented Invisible DDT (IDDT) at BBN wherein the debugging program apparently was not sharing computer memory with the program it was being used to debug.

29. Dan Murphy. "The Beginnings of TECO," *IEEE Annals of the History of Computing*, vol. 31, no. 2, 2009, pp. 110–115.

30. R. J. McQuillin, A Complex Algebraic Compiler for the Royal Precisions LGP-30 Digital Computer, BBN Report 756, June 30, 1960.

31. BBN Reports 755, 759 and 760, authored by Ed Kerwin, an early and longtime BBN employee on the acoustics (and vibration control) side of the house

32. For a copy of the LGP-30 manual, see `http://ed-thelen.org/comp-hist/lgp-30-man.html`

33. Sources of information for this section were Ed Fredkin (phone conversation of September 12, 2002, and follow-up e-mails); Rollo Silver (e-mail of October 24, 2002); Bill Fletcher (e-mail of February 23, 2003); Buzz Bloom (e-mail of March 11, 2003); and the DECAL-BBN Programming Manual.[35]

34. Rollo Silver's trace program was a hack that could trace the operation of a program in the forward or reverse directions. According to Fredkin, it went in the reverse direction "from step n by saving the initial conditions and running forwards n-1 steps." Silver says, "It was fast since it used PDP-1 instructions to help itself."

35. R. J. McQuillin. DECAL-BBN Programming Manual, BBN Report 1051, September 1, 1963; also published as DECUS 39.

36. Licklider's Libraries of the Future project (see Chapter 3) made use of DECAL.

37. The aforementioned programming manual and a technical manual: R. J. McQuillin, DECAL-BBN Technical Manual, BBN Report 1052, November 1, 1963.

38. Sources of information for this section were John Barnaby (telephone conversation of April 26, 2004); Norm Doelling (telephone conversation of September 26, 2002); Wally Feurzeig (emails of September 4 and 11, 2002, June 2, 2003, and April 26, 2004); Bill Fletcher (e-mail of October 21, 2002); Ed Fredkin (e-mail of September 12, 2002); Bob Morgan (e-mails of April 3 and April 15, 2003, and June 10, 2003); Dan Murphy (e-mail of February 25, 2003); Fred Webb (e-mail of April 26, 2004); and Paul Wexelblat (email of April 26, 2004).

39. JOSS — Johnniac Open Shop System — developed in 1963–1965 by Cliff Shaw at RAND Corporation to run on the Johnniac early time-shared computer to provide an interactive programming environment to users.

40. C. Robert Morgan, *Building an Optimizing Compiler*, Digital Press, 1998.

41. MUMPS: The MGH Utility Multi-Programming Systems. Hospital Computer Project, Laboratory of Computer Science, Massachusetts General Hospital, Boston, MA, January 10, 1968.

42. Neil Pappalardo of MGH led the implementation of MUMPS; see `http://en.wikipedia.org/wiki/MUMPS`

43. Virginia M. Dominick. STRCOMP and ISRCOMP: A User's Manual. Bolt Beranek and Newman Inc., Cambridge, MA, April 1969.

44. Barnaby is the same John or Rob Barnaby mentioned in Chapter 21 for his temper and impact on the personal computer world.

45. Randy's implementation was a hack by the definition given in my companion chapter (Chapter 21). In a less useful hack, Fred Webb implemented LISP in STRCOMP, just to show it could be done. A LISP CONS took about 10 minutes.

46. Also see Anit Chakraborty, Randy Graebner, and Tom Stocky, "Logo: A Project History," December 10, 1999, `http://web.mit.edu/6.933/www/LogoFinalPaper.pdf`

47. Later, Bernie Cosell implemented a signal-processing version of MUMPS called BUMPS: BUMPS User's Manual, Bolt Beranek and Newman Inc., Cambridge, MA. February 1969.

48. When I have told people over the years that I worked for BBN, on a surprising number of occasions they have replied, "Oh, I know BBN. I used its TELCOMP service in the 1960s. It was wonderful." And coeditor Ray Nickerson reports, "After coming to BBN in the mid-1960s, I taught an introductory course on computers at Tufts University (under a Tufts contract with BBN) for several years. A feature of the course during the latter part of that time was access from campus to TELCOMP via teletypewriters and phone lines, giving students hands-on programming experience with a time-shared system — an unusual, if not unique, experience in a liberal arts college at the time."

49. BBN was within Tip O'Neil's congressional district. Fantasia's bar was across the parking lot from BBN, and lots of the Democratic Party activity happened there as well as other goings-on.

Part II
Culture, Business, and Management

The second part of this volume deals with BBN's culture, business approaches, and management. In Chapter 5 Dave Walden describes BBN's culture. In Chapter 6 Steve Levy itemizes BBN's technology transfer activities over a fifty-year period. In Chapter 7 Frank Heart describes his view of managing an R&D group in the BBN environment.

Chapter 5

The Way We Were: Aspects of the Culture of BBN

David Walden

The BBN culture had its origins in the academic background of its 1948 founders and in Leo Beranek's approach to business management and entrepreneurship (see Chapter 1). The culture got a significant boost during J. C. R. Licklider's 1957–1962 tenure (Chapter 3) when computer people began joining the company. This chapter attempts to give a fairly comprehensive picture of BBN's quasi-university, quasi-business cultural flavor while recognizing that we can barely touch on all the activities and interests of the many BBNers over the decades.

A chapter in Katie Hafner and Matthew Lyon's best-selling book *Where Wizards Stay Up Late: The Origins of the Internet*[1] is titled "The Third University." The phrase was a reference to Bolt Beranek and Newman Inc.'s (BBN's) location in Cambridge, Massachusetts, and to BBN's having a culture closer in many ways to those of Harvard and MIT than to that of a typical company. It's not clear whether anyone other than people from BBN thought of BBN as "the third university." However, many BBNers thought of themselves as being in the same league as well-known Harvard and MIT professors, and the company used its unusual culture as an aid to recruiting talented staff members and to getting interesting research contracts.

5.1 The third university

BBN was founded by university professors. Although it started as an acoustics consulting company, its main activities in the computer area were research and development. Much of the computer research done at BBN was similar to research being done in university settings. The atmosphere established by the founders was very much that of a university as well; BBN lacked a student body and walls of ivy, but it fostered a spirit of inquiry and research that one expects to find in university contexts.

John Swets remembers,[2]

> [A] distinctive research culture that emerged in MIT's Rad Lab was carried to BBN.
>
> Very importantly, BBN rewarded disciplinary or professional identification and achievement as much or more than institutional loyalty. It didn't mind being a home for entrepreneurs, who thought of BBN as a company that would provide background support in contracts, accounting, facilities, drafting and printing, purchasing, legal issues, etc. and otherwise stay out of the way.
>
> I knew, for example, that I could use BBN's PDP-1, etc., to get and perform on contracts I initiated and hire assistants into BBN's existing departments where someone would provide unobtrusive housekeeping. BBN allowed contractors to come in and discuss ideas with individual researchers who had no management status in the company.

> Leo would hire people who he knew would do their own thing as a matter of fundamental principle, and it never occurred to Licklider to keep an eye on most of the people he hired.

In his chapter on the early years of BBN (Chapter 1), Leo Beranek describes his collegial management style with an emphasis on professional development. He says,

> Overall, my management style was to work with the staff whenever possible, to treat the staff as equals, and to make them aware that BBN was a highly professional organization. Licklider exemplified this same style. I held weekly meetings with senior members of the staff to learn what needed to be done to improve our operations. In writing, I encouraged our staff to become members in appropriate technical societies and to write papers for publication. BBN authorized attendance at any technical meeting where an employee was to present a paper, provided the division head said the paper was first class. If no paper was being presented, attendance at one meeting a year was automatic. Attendance at an additional meeting was approved if there was to be a specially informative symposium. This attitude then carried over into the computer work that followed, . . .

Internet pioneer Robert Kahn, who was part of BBN's ARPANET development team before he went to ARPA, has said,[3]

> BBN was a kind of hybrid of Harvard and MIT in the sense that most of the people there were either faculty or former faculty at either Harvard or MIT. If you've ever spent any time at either of those places, you would know what a unique kind of organization BBN was. A lot of students at those places spent time at BBN. It was kind of like a super hyped-up version of the union of the two, except that you didn't have to worry about classes and teaching. You could just focus on research. It was sort of the cognac of the research business, very distilled. The culture of BBN at the time was to do interesting things and move on to the next interesting thing. There was more incentive to come up with interesting ideas and explore them than to try to capitalize on them once they had been developed.

BBN provided a support structure for self-motivated researchers who were able and willing to find sponsors for the research they wanted to do. The work was not directed in a top-down fashion; within broad limits, senior researchers were free to pursue their own research interests, if they could find the necessary financial backing. Small projects, involving only one person or a very few people, were allowed, as were large projects requiring multidisciplinary teams.

Companies where individual employees seek their own work and remain employed as long as they are sufficiently chargeable are not unusual. Law firms and management consulting firms often operate in this manner. However, BBN people often sought research contracts through which they could advance their own research interests, much like many university researchers. (Of course, in some cases both BBN employees and university professors did consulting jobs on which they provided expert assistance rather than doing original research.)

No one at BBN had a formal guarantee of long-term employment (there was no concept of tenure), but there was a sense of commitment that worked both ways — company-to-employee and employee-to-company. John Swets suggests[4] that this stability may have resulted partly from the ability of many BBNers to react and retool quickly in response to changes in the research support environment. In any case, during the first 40 years of BBN's existence people were not fired from BBN without exceptionally good cause.[5] Occasionally it became clear that this or that person was not a good match for the BBN environment — the need to maintain a reasonably high level of chargeability, to get work done on schedule and within budget, to demonstrate reasonable competence, and so forth. In such instances, the company typically gave an indefinite amount of time to leave — perhaps months — and sometimes helped them to find more suitable employment. At minimum, the company did not broadcast that an employee had been encouraged to find a more appropriate situation, and the employee was free to tell his or her own story about why he or she no longer wanted to be at BBN.

Being like a university was not advantageous in all respects. Because BBNers were conducting similar research in many cases, to that done at universities, they often found themselves competing with university teams on proposals. In part because the company lacked graduate-student labor, and in part because it was a profit-seeking enterprise, BBN's unit labor costs were generally higher than those of universities. This was a bidding disadvantage that kept pressure on the organization to produce proposals that could win bids on technical merit when competing against lower-cost bids. Thus, although BBN was a profit-seeking company, it did take no-fee contracts or grants when support for a desired project could not otherwise be obtained.

Curiously, BBNers were sometimes more likely to work together on projects, share research, and so on, than faculty members within universities. For up-and-coming university faculty members, tenure is often the primary goal, and this is an individual reward; in contrast, BBNers tended to succeed by working together in teams.

5.2 Recruiting, developing, and keeping employees

BBN has always sought to recruit very bright and highly regarded technical talent. In Leo Beranek's chapter (Chapter 1) he says,

> Above all, I insisted that the motto of the company be, "Each new person hired should raise the average level of competence of the firm." This became an operating creed that kept us from hiring anyone who we believed was not as smart as ourselves.

Licklider also espoused the principle of hiring only people who raised the average level of intelligence, according to Ed Fredkin;[6] and John Senders quoted[7] Licklider as saying that the single operating rule of BBN was that if you met someone as smart as yourself you hired him/her.[8] Intelligence was not all that BBN looked for in potential hires. Integrity and articulateness (in writing and orally) were valued. Job hoppers were avoided.

More generally, when an exceptionally smart (by BBN's high standards) person was available to be hired, BBN often hired that person whether or not a specific project needed staff and even in the face of tight economic times. Since exceptionally smart people were key to the company's long-term success, they had to be hired when they were available — there wouldn't be a second chance later.

Naturally, there was competition from other companies, and sometimes from universities, in the quest to hire these excellent people and keep them for the long term. In Chapter 1, Leo Beranek also describes three devices that the early top management of BBN created to help recruit and keep such in-demand people: the k-factor profit- (and loss-) sharing plan, a stock purchase plan, and a technical promotion ladder (consultant, scientist, engineer, senior, principal, chief) with salaries somewhat matched to the management ladder.

≈ ≈

Hiring people whom you personally know to be good or who are vouched for by people you respect is a time-honored approach to recruiting. From the beginning, BBN people used their connections with some of the major local universities (e.g., MIT, Harvard) to help recruit additions to the BBN staff.

Many BBN people were recently from universities (having been either faculty or students) and already knew good people there. One excellent example (described in John Swets's chapter [Chapter 3] and in a book by Waldrop[9]) was Licklider's hiring his "dream team" of psychologists and psychoacousticians to join him at BBN and then half staffing his Libraries of the Future project with MIT graduate students he knew.

Some people continued to teach university courses while working at BBN, either regularly or from time to time, and met good students in their courses. For instance, for years Dick Pew organized and taught in a summer session at the University of Michigan. For several years, Severo Ornstein taught a hardware design course at Harvard[10] and out of this he noticed and recruited (or his recruits recruited) person after person who joined BBN's ARPANET team: Ben Barker, John McQuillan, Mike Kraley, Marty Thrope, Joel Levin, and others. When John McQuillan and I taught the first course anywhere on practical packet-switching network design, the students surprised us by reserving the Harvard-Radcliffe bus for the last day of class and demanding that we bring them to BBN to show them "packet switching really happening" and to tell them about job possibilities; as a result of their field trip, we hired four people from our dozen-or-so person graduate seminar, including Eric Roberts, later a professor of computer science and a dean at Stanford University.

MIT Professor Ken Stevens consulted part-time at BBN for decades, and BBN hired many of his best students over the years, including John Makhoul (Chapter 14), Jerry Wolf, and Ray Tomlinson.

In my view, top-flight R&D groups, like professional sports teams, are best built through the college "draft." Sometimes you hire great people who have worked at other places, but all too often there is something wrong with a person who comes to you from another company — if the person was really so great, why did the other company let them get away? Furthermore, perhaps only one person in 10 or 20 who you think has potential to be a superstar actually turns out to be one.[11] However, you can't afford to hire 10 or 20 people already working for other companies who have already shown superstar capability, whereas you can afford to hire 10 or 20 new graduates with their lesser salaries. Thus, college recruiting was always a key endeavor for BBN.

BBN was in a particularly fortunate position with regard to the many Boston-area educational institutions. We often got recommendations from faculty members we knew; in addition, we often were able to hire people summers or part-time in the years before they graduated. In this way we got a relatively free look at their capabilities, and we could begin developing attractive permanent situations for the summer and part-time hires who showed the most promise. Once they actually graduated, inertia was often on our side. In addition to having a competitive financial offer from BBN, new graduates could keep their same boyfriend or girlfriend, keep their same student apartment for a while, and keep bicycling the same familiar route to BBN.[12]

Of course, in the computer area, you can't just recruit from MIT and Harvard. Consequently, BBN developed a recruiting program at several top computer science schools and schools that excelled at developing practical engineers. During my time as general manager and president of BBN's R&D activities, I personally went on college recruiting visits, where I enjoyed meeting the superbright about-to-graduate students and telling them that my most important job was finding good new people for BBN.

≈ ≈

Like a Bell Labs or Watson Research, BBN always had lots of technical staff members with advanced degrees, at least compared with the average company in the computer world. On the other hand, there were always a few members of the technical staff without even a bachelor's degree.

BBN embraced all the normal staff development activities that many companies have: tuition reimbursement for courses taken toward a degree, a variety of introductory and project management and skill-training courses, periodic performance reviews, and the like. However, BBN also has a more extensive than typical development program known as the Science Development Program.

According to John Swets's memory,[13] then president Sam Labate conceived the Science Development Program in the spring of 1975. Six months earlier, John had moved from being general manager of BBN to being principal scientist. He and Jim Barger were named chief scientists (for information and for physical sciences, respectively) with responsibility for running SDP. In 1982, at the time I was appointed general manager of BBN's R&D activities, Steve Levy appointed Ray Nickerson to run the SDP program (relieving John and Jim from duty). When Ray retired from BBN in 1991, John Swets again led the SDP until his own retirement in 1998, at which time John Makhoul took on responsibility for SDP.

Each year an annual report gives an account of the SDP's activities over the year. The report from 1987 had the following introduction, which sketched the purpose and activities of the SDP:

> Research, development, and consulting have always been the principal work of BBN's traditional business unit. The quality of this work brought BBN widespread recognition as an innovative leader in its areas of technical specialization. Innovation begins, however, with the capability of the technical staff, so BBN . . . considers it essential that scientists and engineers have continuing opportunities to enhance their professional development. The company established the Science Development Program (SDP) to promote scientific and professional staff development by providing financial support that allows staff members to attend conferences and make presentations, publish papers and books, and serve on professional committees or advisory councils. Other components of the SDP include: educational activities, special interest seminar series, guest lecturer series, film series, visiting scientist program, sabbatical program.

A typical report — from 1987 — included the items shown in Table 5.1.[14]

≈ ≈

Another component of BBN's ability to attract and keep first-rate scientists and engineers was toleration of individual differences — letting people be themselves without regimentation.

Until the later 1960s some employees brought their beloved dogs to work with them. Offices were appointed to suit the occupants' tastes, with physical plant people available to install, for instance, wall-hung bookshelves where an employee wanted them. Many offices had nonstandard (more comfortable) chairs in them. Some office furnishings extended into the public halls.[15]

There was no dress code. Informal clothes were the norm unless individuals (including many senior managers) chose otherwise. BBN informal was not like relatively sharp looking "casual Friday" attire at companies where men otherwise wear suits and ties. Unironed shirts and jeans that needed washing were common. In summer months some people wore shorts. Boat shoes without socks also were common, and Bob Brooks was legendary for his years of going barefoot around BBN, summer and winter.

Typically employees wore more businesslike dress when they visited customers or when customers came to visit them at BBN.[16] However, there were exceptions. Katie

Table 5.1 Content of a typical SDP annual report.

- List with brief biographies of the chief, principal, and division scientists, engineers, and consultants
- Profiles of several notable scientists, engineers, or consultants
- A sketch of educational resources available to the staff
- Lists of the seminar and film series
    1. List of the several speakers in the guest lecturer series (these speakers were typically world-class talents in their respective fields)[a]
    2. List of the seminar series and speakers and topics within each seminar series (in 1987 there were six seminar series with a total of 124 different presentations — essentially one every other work day)[b]
    3. List of the technical films and videos shown in the film series
- Biographies and summaries of the work done by scientists who spent sabbaticals at BBN[c]
- Descriptions of the work being done by BBN scientists doing sabbaticals elsewhere
- Lists of academic institution activities BBNers participated in (see page 80)
- List of awards and honors given to BBNers
- Representative list of publications of BBNers
- Additions to the BBN authors' bookshelf in the BBN library
- Update of the time line (since 1948) of notable technical achievements

[a]These companywide talks sponsored by the Science Development Program strengthened the ties between BBN and universities by exposing leading (and other) university researchers to BBN and vice versa. Guest lecturers through the mid-1990s included Nobel laureates Walter Gilbert, Richard Feynman, Sheldon Glashow, Herbert Simon, and Kenneth Wilson, and other notables such as Stephen J. Gould, Victor Weisskopf, Lewis Branscomb, Benoit Mandelbrot, Donald Michie, Persi Diaconis, Lynn Margolis, and Philip Morrison.

[b]The frequent technical seminars typically had invited guests who presented papers, although sometimes BBNers were featured. One seminar (on cognition) was held more or less monthly for several years and was regularly attended not only by BBNers but by faculty and students from several of the local universities — MIT, Harvard, Brandeis, BU, etc.

[c]BBN received many requests from academics to spend sabbatical time at the company. Often these overtures came from colleagues who wished to spend their sabbatical working with specific BBNers or in particular departments. BBN honored these requests when it could, and had numerous sabbatical visitors over the years.

Hafner's popular history-of-the-Internet book[1] recounted the incident when Frank Heart, Bob Kahn, Severo Ornstein, and Will Crowther were called to visit ARPA in the final stages of defending BBN's ARPANET proposal — and Crowther ignored Heart's suggestion that he should wear something other than his customary sneakers. Of course, BBN won the ARPANET contract anyway; customers who appreciated BBN's technical capabilities were not too worried about how BBNers dressed. In another example, a program manager at ARPA scheduling a visit to BBN said, "I know that you are not dressed up when I am not there — so rather than you dressing up to host my visit, how about if I dress down to visit you?"

BBN also has always been good at accommodating employees' special circumstances. On occasions when a valued employee could not live near a BBN office, he or she was allowed to telecommute. Starting before telecommuting became well known, Craig Partridge, co-author of one of our chapters (Chapter 17) has not lived near the BBN office of the research group he was in. Also, on many occasions employees have obtained advanced degrees while continuing to work full-time at BBN, with BBN sometimes pro-

Figure 5.1. Bob Brooks has been often mentioned in stories of BBN but seldom shown. He had the same wash pants, work shirt, and absence of shoes whether he was outside or at work inside at BBN. (Photo courtesy of Bob and Hester Brooks.)

viding a thesis topic. For instance, John McQuillan's Harvard PhD thesis was on network routing,[17] with the ARPANET (for which he was at the time the lead programmer) as his experimental test bed; this work ultimately led to an ARPA contract under which John developed the routing techniques[18] that are now used throughout the world under the name of OSPF.

≈ ≈

Although the "third university" story was relevant to top technical people obtaining their own funding to do research with a small cadre of research collaborators and assistance, over the years a majority of BBN's work was development or even system building. Development and systems work require organization and staffing more like those in a traditional product company: a few senior technical people, a few technical and project managers, and quite a few (perhaps dozens of) journeymen workers to carry out lots of implementation tasks, to maintain systems, and so on. The managers of groups doing this sort of work were not so interested in hiring people with excessively academic inclinations. These managers sought people who were more entrepreneurial, more do-what-the-customer-needs oriented, and interested in profitability and business growth. There were some under-the-surface resentments between the more universitylike groups, who thought BBN was investing too much money in new ventures, and the groups more oriented to business growth, who thought BBN was spending too much money on sabbaticals and other academic trappings. There were also dissatisfactions and inconsistencies, with (a) some more research-oriented people wanting financial participation in the entrepreneurial ventures, but with (b) others from these same groups resenting control of the technologies being moved to a more entrepreneurial group.

Of course, at all levels you want people who are relatively bright and relatively talented. Sometimes lower-level people could be the superstars of the future who were still gaining experience. More often, however, the lower-level people (and some of the top-level people) were not superstars; typically they were above average, but in some cases they were below average (and perhaps not just at BBN but in the world at large). The myth aspect of the BBN "third university" story (all about highly talented people doing what they wanted) could be a problem with the more average employees. A manager might be willing to put up with prima donna behavior from a superstar (although in most cases superstars didn't have outsized egos or behavior), but he or she would be less willing to put up with prima donna behavior from superstar wannabes (and BBN attracted some of these). Various managers just wanted most of their people to do what they were asked to do without arguing back very often (for example, to implement what the customer asked for, fill in time sheets as specified, accept the way accounting is conventionally done, etc.). While it was an exaggeration, I used to go home in the evening after a day trying to manage a BBN division that I was heading and say to my wife, "I wish that just once someone who works in my division would do what I ask without me having to beg." Another senior manager used to say, "I don't want people for whom work is their hobby and their avocations are what they mainly care about."

At a university, the department head position is often one that rotates among top faculty members, each of whom does it for a few years before escaping back to his or her research career. Thus, at a university, the department head is likely to be in the same technical league as the other faculty members. As a BBN technical manager, however, there was a good chance that you were responsible for people technically (and maybe absolutely) smarter and more talented than yourself. Management positions were not subject to rotation; rather, people tended to separate permanently onto the management or technical paths. (There were some notable exceptions: top managers gave up the management path and returned to technical work, as in the cases of John Swets and Jim Barger.) Thus, the technical talents of some people on the management path paled in comparison with those of the people on the technical path. This is not different from the way many companies operate. However, at BBN, with its many top-rank people and its story of individual freedom that more average employees embraced to a fault, managing technical people could be quite a trial.

Also, the top technical people at BBN were so valuable and so highly regarded that they were sometimes treated better than some managers (higher pay, sabbaticals not available to managers regardless of the extent of their contribution, etc.). This could take some getting used to. You had to tell yourself that it really was important for your own good that there were people working for you who were so much smarter than you, so independent, and treated better.

5.3 University ties

Because BBN was something of a cross between a university science and engineering faculty and a company, it has always had extensive university ties.

As mentioned earlier, many of the BBN staff had been university professors before joining BBN and when senior people left BBN it was often for a university position; some gave courses at universities as BBN employees. Many BBNers were active in professional associations and societies as officers, committee members or chairs, and conference participants. Many published in the technical literature, gave talks at university seminars, and served on PhD committees. As a consequence of such activities and numerous collegial relationships between BBNers and university faculty, the company's visibility at universities was high, and professors often recommended outstanding students for employment. Most of the BBN-university relationships were informal and ongoing; on numerous occasions, however, BBN formally teamed with university groups to undertake specific projects. Sometimes the resulting consortia lasted for several years.

In 1985–86 Ray Nickerson wrote a summary of university ties as of that time and came up with the not atypical (for BBN) list shown in Table 5.2 (and despite its length, he may have missed some activities).

≈ ≈

BBN even started its own graduate school at one point — the Program for Advanced Study — in which BBN scientists and engineers taught state-of-the-art engineering and science in public courses (including people from competitors). Of this, Steve Levy says,[19]

> I believe that the Program for Advanced Study (PAS) was created by BBN in 1964 to provide busy professionals an opportunity to efficiently keep up to date with an ever-expanding body of scientific and engineering knowledge without having to leave their full-time jobs and go back to a university. As such, it may have been one of the earliest examples of a commercial program of continuing professional education.
>
> In 1966, Ira Dyer was President of PAS and Bob Johnson and John Bennett were Associate Directors. Courses were given in several cities around the country, often at the facilities of a host company.[20]

PAS continued for a few years, with many BBNers teaching courses in it. A typical course might be four full days long — a two-day Friday–Saturday sessions, a few weeks for the participants to do homework, and a concluding Friday–Saturday session. Friday–Saturday pairs of days were used so the students in a course and their companies shared the time to actually attend course sessions. The courses were serious advanced technical courses. There were more courses from the acoustics side of the company than from the computer side of the company.

In time the program struggled, its management changed (Walter Kolton led the effort for a while), and finally it was shut down. Nonetheless, PAS was indicative of the ongoing BBN struggle to publish and teach what employees knew (at a university) rather than keeping quiet and trying to hold onto a competitive advantage.

5.4 Entrepreneurial urges

There has always been an entrepreneurial aspect to BBN, in addition to the "third university" culture. Leo and Dick started the company to pursue opportunities that didn't fit in the MIT environment. While they staffed and managed BBN more like a

Table 5.2 Example list of university ties.

- BBN collaborations with universities on grants and contracts — 14 projects, 13 universities
- University faculty members consulting or working parttime at BBN — 17 individuals, 13 universities
- Summer and part-time student employees — from 24 universities
- BBN employees helping teach university colloquia, seminars, and short courses — 20 employees, 47 universities
- BBN employees serving as university adjunct faculty or research affiliates — 9 employees, 8 universities
- BBN employees serving as PhD dissertation advisors or on PhD advisory committees — 8 employees and universities
- BBN employees serving as chairs of National Academy of Sciences and National Research Council committees — 3 employees and committees
- BBN employees serving on advisory boards, panels, and task forces for organizations that support academic research–16 employees, 24 boards
- BBN employees serving as proposal reviewers for agencies that fund academic research — 23 employees, 7 agencies
- BBN employees serving as officers and committee members or chairs for professional societies with significant academic membership — 18 employees, numerous societies
- BBN employees serving as editors or on editorial boards for academic or scientific journals — 17 employees and journals
- BBN operation of major national resources serving the academic community: 3 instances
- Sponsorship of seminars open to faculty and students of local universities: 3 seminar series (some of which had been active for years), 100 seminars over the course of a year
- Sponsorship of visiting scientists: 3
- Sponsorship of prolonged visits by BBN employees at universities: in a typical year, a couple of BBN principal scientists were on sabbatical at a university
- Sponsorship of employee continuing education: many employees taking courses at universities, many university courses available at BBN via satellite, several videotape courses offered at BBN each year

university department, they pursued myriad business opportunities. In Chapter 6, Steve Levy describes many new business thrusts over the years, starting with an effort as early as the 1950s to license new technology. Leo was so interested in business that in the early 1970s he left BBN entirely to pursue an opportunity in television station ownership.[21]

In addition to the many start-up activities at BBN over the years, many traditional contract R&D activities were involved in development and systems work versus research. While managers of these development and systems activities did not hold having a PhD against someone (particularly in Frank Heart's Computer Systems Division), they hired anyone who could do the work. As Paul Castleman reports (Chapter 12), in the medical systems development group they sought PhDs only in the application area, such as pharmacology or genetics.

There was an ongoing undercurrent (or sometimes visible current) of tension between the more third-university technical activities and the activities more oriented toward delivery of working systems. Engineers and scientists wanted both the joy and freedom of working for BBN and the financial rewards from a start-up, which tended to drive BBN in the direction of having many internal start-ups. There was an ongoing struggle to maintain the traditional BBN environment (which many BBN *researchers* especially cherished and most *engineers* liked) while pursuing business opportunities (which some *researchers* resented and which the involved *engineers* often thought were not being pursued hard enough). In fact, the substantial migration of BBN people to Xerox PARC occurred partly because they were seeking a place where researchers did not have to seek funding themselves; this was also always Frank Heart's goal (Chapter 7).

Also, many of BBN's development and systems projects happened partly because the time was ripe for them in terms of the technological state of the art and economics. Thus, project engineers could see possibilities of becoming a company like Sun or Cisco became. Also, people who became technical managers tended to be entrepreneurs. However, curiously, in retrospect there is a tendency for longtime BBN people to blame management for spending all that money and not bringing in people who were "real business people" or "real marketing people" to lead the activities. It is ironic that some of the engineers who were given the opportunity to be senior managers of these business activities or to be key contributors (all hoping for big success of one sort or another) make some of the most negative statements about BBN's management's not having had people qualified to pursue such businesses.

Personally, I think we were relatively successful on many occasions, and I think there is little point in speculating why we didn't become Sun or Cisco — no more point than in trying to figure out what combination of capability and luck made Sun and Cisco Sun and Cisco. In the end, BBN's contribution was to be a great place to work, a place that developed a lot of people, moved lots of technologies ahead, and spun off many other activities.

5.5 Extracurricular activities

A description of BBN would not be complete without discussion of the many aspects of the day-to-day life that made it not only an intellectually exciting place to work, but a playful place as well. I can categorize these extracurricular activities into two main categories, hacking and recreations.

Computer Hacking

From the time of the arrival of the first PDP-1, computer hacking was a significant activity at BBN. By hacking, I don't mean illegally breaking into computer systems, distributing viruses, and so on, which is what "hacking" means to many people today. Rather, I use the original meanings from pages 216 and 218 of *The New Hacker's Dictionary*:[22] *hack* (n: a quick job that produces what is needed) and *hacker* (n: a person who enjoys exploring the details of programmable systems and how to stretch their capabilities). Surely hundreds or thousands of hacks were done at BBN over the years.

There are several motivations for hacks. Some hacks were work related; for instance, the creation of a new tool (that would be useful for a specific project or useful more generally) that was not part of any project or annual tool-building plan. Some hacks were personal education projects. Some had to do with the non-BBN interests of BBN people.

The original BBN hackers were Licklider and Fredkin, whose exploits are described in several other papers in this book (particularly in Chapter 4). Fredkin's activities to design new instructions for the original BBN PDP-1 and to write utility software for it were certainly hacks, and Fredkin had a hacker's mentality — indeed, for all his productivity, he had a hard time doing assigned work. Still early in the BBN computer era, the development of a BBN version of JOSS that led to BBN's TELCOMP system was a bet-you-a-dinner hack done by Bob Morgan, Dave Walton, Steve Weiss (see Chapter 4 for the complete story).

Later hacks (both described in Hafner's book[1]) were Bernie Cosell's creation of utility software to aid the ARPANET software effort and Ray Tomlinson's demonstration of the first networked e-mail system. Another hack worth mentioning here is the software developed by Will Crowther and other BBNers for creating the maps related to the exploration of Mammouth Cave.[23]

Recreations

In many companies groups of people jog together before work, play softball after work, go skiing on weekends, or play bridge at lunch hour. Such activities went on at BBN. However, at BBN the amount of time spent and the breadth of recreational activities were sometimes more comparable to student clubs and activities at a university. Also, as at a university, some of these activities were partially supported by the company. Some of these activities went on for years (decades for one of the lunchtime bridge games); others were rather intense fads.

Sports activities at BBN included a dart phase; Ping-Pong; bike riding; basketball; softball; volleyball; Tai Chi; a fencing phase; a tumbling phase; an annual open golf tournament (rain or shine, for players of all levels including complete novices); bowling leagues (one that went on for decades); a juggling phase (it spread to other early Internet sites; and for several years BBN seemed like the de facto administrative headquarters of the International Jugglers Association; see Figure 5.2); and sailing (including a fairly formal navigation course and more than one racing team). There was also a phase in which people were learned to fly (and jointly bought) airplanes, if that can be considered a sport.

There is a stereotype that people who are good at math like music. That certainly seemed true at BBN, where music was another significant area of recreational activity. Many individuals took work or thought breaks playing instruments in their offices. There was a company-provided piano around BBN for years,[24] and other individuals took breaks playing this piano. Several BBN employees were essentially professional musicians in their non-day job. From 1980 to 1985 or 1986, the BBN Singers practiced and performed. In 1984 Ray Nickerson organized and MC'd an evening concert of BBN musicians.

Puzzles and games were another big area of recreation, sometimes undertaken at such an intense level and involving so many people that an outside observer might have concluded that the game or puzzle was part of official work. There was a lunchtime bridge game that went on for decades. There was a Klaberjass (a card game) phase and a Twixt (sort of like Chinese checkers) phase. The 1972 Fischer-Spassky world championship chess match led to a major postal chess phase at BBN. Car rallies by some BBNers led to participation by more BBNers in the annual St. Valentine's Day Massacre map rally,[25] which seemed like the main business of the Computer Systems Division for several weeks a year.[26] The Rubik's Cube fad was a near-constant "work" project at BBN, with lots of group theory analysis. When a collection of BBNers learned about Dungeons and Dragons, the dungeon master created a game that was particularly detailed, went on for a year, and concluded with a 100-page "final report;"[27] Will Crowther, a participant in Mirkwood Tales, soon after created the first computer adventure game.[28]

Figure 5.2. Dave Walden. "I was an enthusiastic novice, and my enthusiasm spread through BBN and out into the early Internet community." (Photo by Alex McKenzie.)

5.6 Student protests

There is always a lot of left-of-center political activity in Cambridge, emanating from Harvard and MIT and from the city itself (not for nothing is Cambridge known to conservatives as the People's Republic of Cambridge). Thus, BBNers work and many live in the liberal Cambridge environment, and many BBNers are liberal thinkers (and doers).[29]

Much of BBN's work has always been for the Defense Department. Undoubtedly a majority of employees working on DoD contracts were not troubled by their involvement.[30] However, working on such contracts, or even working in a company that took such contracts, was troubling to some employees. Nonetheless, the work was often fascinating, and many employees who had antimilitary leanings came to some personal rationalization about working for DoD (although from time to time an employee left the company because he or she could no longer justify doing the DoD work). Thus, there was also always a small undercurrent of dissatisfaction with things military and BBN's role in them, not unlike the dissatisfactions on the same front in many universities.

At various times, BBN's business faced offical threats from the city of Cambridge.

For instance, on one occasion there there was a push to make it illegal to do nuclear work of any type in Cambridge. This would have been a problem for BBN, which worked substantially in areas relating to nuclear submarines and to a small extent with nuclear power plants.[31]

One protest came right to BBN. It must have been sometime in the 1970s, when there had been a series of antiwar or antinuclear protests at the universities in Cambridge. One day we heard that the protesters would be coming to BBN. At the scheduled hour on the scheduled date, the protesters arrived in front of BBN's 50 Moulton Street (headquarters) building. A number of us BBNers were standing outside on Moulton Street watching to see what would happen. I don't remember Leo Beranek's being there, but I do remember that Dick Bolt was out in front of the building. Once the protesters appeared ready to "do their thing" (as we would say then), they were perhaps surprised to see a group of 8 or 10 BBNers come out the double front doors of 50 Moulton Street to join the protesters and to hand out their own little flyer welcoming the protestors.[32] I am sure that some of the BBNers who joined the protest worked on projects inconsistent with the aims of the antiwar protest (e.g., Herb Fox who headed the BBN protest committee, Bernie Cosell, etc.). Next came the most surprising action of the day. Board member Jordan Baruch, speaking for the company, invited all of the protesters to come onto BBN's little lawn in front of 50 Moulton Street, noting that once people were on BBN's private lawn they couldn't be hassled by the police for anything like obstruction of a public way. And the protesters mostly did come onto the lawn. With BBNers participating in the protest and the protesters safe from the police on BBN's lawn, there was no violent interchange between protesters and police (I think things may have been so peaceful that the police were never called; in any case, they certainly were not called by BBN). With no violence happening, TV crews never came, and the outside protesters soon departed. It was all pretty much a big "NO OP," as a BBN computer person might say. It was also pretty smart thinking by BBN's top management. Bernie Cosell remembers it as ". . . a really amazing (and a bit scary) experience."[33]

A particularly rude example of protest by a BBN person involved founder Dick Bolt. Bolt had been active in the American Association for the Advancement of Science (AAAS) for many years. In 1970–1971, renowned nuclear scientist Glenn Seaborg was a candidate to be the next president of the AAAS, and Dick Bolt was also on the ballot for the position. At the AAAS meeting in Boston that year, a BBN employee publicly protested Seaborg's candidacy, presumably objecting to his involvement in nuclear research. Naturally, this was greatly embarrassing to Dick Bolt — to have an employee of his perhaps appearing to be campaigning in public to undercut Seaborg's candidacy in favor of Bolt's.

Finally, BBN had its own lampoon efforts. One of BBN's traditions was the Sturdleigh letters (although everyone knew that Ed Kerwin was the real BBNer behind the Sturdleigh nom de plume) that were published for years immediately after the annual meeting of shareholders. This admittedly clever report ridiculed everything that happened at the annual meeting and that was said by the top management. Most employees loved the Sturdleigh letters. Top management laughed along: What else could they do without looking petty and inviting more derision by employees?

In January 2006, Dick Horonjeff, a longtime employee of BBN's Los Angeles office, led an effort to collect and scan the complete set of J. C. Sturdleigh "Annual Meeting" newsletters in time for another longtime employee of the office, Colin Gordon, to enjoy them one last time in the days before his death. Dick's effort was made possible through the assistance of Dennis Kerwin, who had the complete set in his library. The complete compendium is posted on the Internet.[34]

In later years, another (more bitter, less clever) report deriding management, the Beanco report, appeared from time to time.

Another tradition until the late 1960s was a Christmas party at which management was roasted. (Later the annual parties took on a different character.)

5.7 Alumni association

Of course, employees who leave any company may later stay in touch with some people they knew from the company. However, the way many BBNers think of their years at BBN and the people they knew at BBN is more like the way many people think of their college years than how people typically think of their work years.

When a longtime BBN person leaves the company, the going-away party announcement inevitably circulates beyond the walls of BBN. A goodly number of ex-BBNers come to the party, to celebrate the person they still consider to be their colleague and to see other friends they haven't seen for a while; the event can take on the feeling of a homecoming.

BBN has been good about giving emeritus status to some long-service employees when they left the company. This honor comes complete with a badge that gives the emeritus employee continued access to BBN resources such as the library.

When Dan Dern left BBN, he organized the xBBN e-mail discussion group.[35] Originally the discussion group was for the purpose of letting ex-BBNers inquire and tip each other off about job opportunities. Over time, however, this list has developed into something more like the alumni news publication of a university. Seven hundred or more ex-BBNers are on the list. When a BBNer or an ex-BBNer is quoted or noted in a newspaper or magazine, pointers are immediately flashed to the xBBN list. Ex-BBNers seek contact information for other ex-BBNers. There are discussions about all manner of things: the best way to handle IRA rollovers or get non-group health insurance, new business ideas, computer configuration problems, social policy and e-mail spam, and so on. When we worked at BBN and wanted information about something, the first people we turned to were the bright set of people with whom we worked, who had amazingly diverse interests and depth of knowledge. Now that we don't work at BBN, many of us still turn to the xBBN list first as the place to get thoughtful insight on any issue.

Some of the long-term extracurricular activities also have continued after people left BBN. For example, people who played lunchtime bridge together for years before they left BBN continued to meet for a weekly game, and the BBN bowling league began to admit ex-BBNers.

For many people, there is a loyalty to and appreciation of BBN, BBNers, and ex-BBNers that is perhaps closer to the relationship a person has for life with college fraternity brothers or sorority sisters than to what many people feel for their ex-company and ex-colleagues. Even stronger, perhaps — since many of us worked and played together for decades rather than just for our years in college.

≈ ≈

What is amazing to me is how much real work we got done and the technical progress we made given how much of our time was spent on other interests. Of course, just as lots of play went on at work, lots of work was done at home (this was one of the benefits of having remote access early on to time-shared interactive computers). I've always said, when people ask me about my work at BBN and my personal activities, "I could never tell the difference."

Acknowledgments

I am grateful to all the people quoted in this paper. Editor Nickerson provided significant text for chapter; however, he demurred at being listed as coauthor. John Swets suggested topics to be described. Specific queries were answered by Bernie Cosell, Billy Diskin, Bobbi Freeman, Dave Getty, Steve Levy, Alex McKenzie, and Ray Nickerson. Review or additional input was provided by Terry Barry, Tim Bergin, Paul Castleman, David Grier, Steve Heinrich, Alex McKenzie, Mike Nacey, Ray Nickerson, John Swets, and the anonymous reviewers of the *IEEE Annals of the History of Computing*.

Notes and References

1. Katie Hafner and Matthew Lyon, *Where Wizards Stay Up Late: The Origins of the Internet*, Touchstone paperback, Simon & Schuster, New York, 1998.

2. E-mail of March 8, 2003.

3. Oral history interview with Robert E. Kahn, interview by William Aspray, in Reston, VA, March 22, 1989. Charles Babbage Institute, University of Minnesota, Minneapolis, call number OH 158.

4. E-mail of June 9, 2003.

5. In the late 1980s and 1990s, BBN carried out economically based reductions-in-force (RIFs), like many other companies at the time.

6. During the communications cited in section 4.1, which starts on page 51.

7. E-mail of April 16, 2003.

8. In later years, Frank Heart often [and loudly] proclaimed his theory of hiring, "Hire the person with the most neurons per cubic centimeter." While not every exceedingly bright person was proportionally productive (or pleasant to be around), if truth be told, Frank's most-neurons theory worked pretty well.

9. M. Mitchell Waldrop, *The Dream Machine: J. C. R. Licklider and the Revolution That Made Computing Personal*, Viking, New York, 2001.

10. Severo M. Ornstein. *Computing in the Middle Ages.* 1stBooks, 2002 `www.1stbooks.com`

11. Of course, a nonsuperstar can still be a valuable employee, so it definitely pays to hire people who look like they have star potential.

12. In addition to hiring people part-time to get a look at them, BBN was a good oasis for many people taking a break from college; notably famous economist Fischer Black (Mehrlin, Perry, *Fischer Black and the Revolutionary Idea of Finance* John Wiley & Sons, Hoboken, NJ, 2005, pp. 40–41, 43, 47, 51) and people on the Hospital Project (page 57).

13. E-mail of May 22, 2003.

14. The lastest SDP report I have seen (2008) includes as impressive a list of activities and accomplishments as ever. Over the years, SDP has also sponsored a number of other items listed in the next section on university ties.

15. Ed Kerwin created an elaborate office tableau on the hall wall opposite his office for "Mr. Small" (although perhaps this belongs in the "neat hack" category described later in this chapter).

16. And many BBN men kept a sports jacket and tie on a hanger on the back of their office doors against a surprise call to meet with someone from outside BBN.

17. John M. McQuillan. Adaptive Routing Algorithms for Distributed Computer Networks. BBN Report 2831, May 1974.

18. John M. McQuillan, Ira Richer, and Eric C. Rosen, "An Overview of the New Routing Algorithm for ARPANET," *Proceedings Sixth Data Communications Symposium*, 1979.

19. E-mail of March 17, 2003.

20. Steve Levy also notes that the formation of PAS was preceded by the formation of the Honor Products Company (HPC) in 1962, which developed training and education programs for businesses and ultimately for consumers.

21. However, Leo never lost his interest in the science and engineering of acoustics. After selling the TV station and spending a number of years on philanthropic opportunities (e.g., as president of the Board of Trustees of the Boston Symphony Orchestra), Leo returned to acoustics consulting. His work on the version of Chapter 1 that was published in the *IEEE Annals of the History of Computing* had to be squeezed among frequent consulting trips he was making — at approximately age 90.

22. Eric S. Raymond, *The New Hacker's Dictionary*, second edition, MIT Press, Cambridge, MA, 1993.

23. Roger W. Brucker and Eichard A. Watson. *The Longest Cave.* Southern Illinois University Press, Carbondale, IL, 1987.

24. See `www.walden-family.com/bbn/TM1296.pdf`

25. See `http://home.earthlink.net/~oldmaltese/Massacre.html`

26. Car rallies were also used for several years as the way to find the Computer Systems Division summer picnic. One year a few members of the division following the car rally instructions on horseback.

27. Eric S. Roberts, *The Mirkwood Tales*, privately printed, 1977.

28. See `http://www.rickadams.org/adventure/a_history.html`

29. It was a big surprise to me when George McGovern lost the presidential election in 49 states — everyone I knew voted for him.

30. And some of us were deferred from military service, e.g., in Vietnam, because we worked on DoD contracts at BBN.

31. On another occasion there was anti-biological-research pressure, but that mattered more to other high-tech companies than to BBN.

32. Bernie Cosell e-mail of May 1, 2010.

33. E-mail of October 31, 2002.

34. `http://www.xbbn.org/files/SPI_AnnMtgs_Ed01.pdf`

35. Later Bernie Cosell, Harry Forsdick, and Tom Fortman helped Dan by providing technical support and maintenance of this list, and eventually they relieved Dan of responsibility for it.

Chapter 6

The History of Technology Transfer at BBN

Stephen Levy

BBN's basic business since its founding has been contract consulting, research, and development. This article describes BBN's activities from 1948 to 1997 to transfer technology and intellectual property from its basic sponsored consulting, research, and development business into a variety of commercial and other products and services.

6.1 Introduction

In this chapter I will describe BBN's efforts to capitalize on technologies emerging from its consulting research and development activities over a fifty year period beginning with its founding in 1948, up to its acquisition by GTE Corporation in 1997. In the hope of presenting a more concise and understandable picture of the varied and sometimes complex technology transfer, commercialization, and related financing activities undertaken by BBN during those fifty years, I have chosen to divide the history into five periods of time:

- 1948–1959 — Early Intellectual Property Transfer Efforts
- 1960–1969 — Licensing, Investments, and "Industrialization"
- 1970–1979 — Computer and Communications Subsidiaries
- 1980–1989 — Rapid Growth and the End of the Cold War
- 1990–1997 — Emergence of the Internet and Acquisition by GTE

Of course, activities begun in one period of time often continued into the following periods, but my intention was to divide BBN's history in such a way as to make clear what I perceive to have been the dominant events that defined that period.

In preparing this paper, I drew extensively on information contained in BBN's Annual Reports from 1961 to 1996, BBN's Initial Public Offering Prospectus dated 1961, various BBN Proxy Statements and Form 10K's, reports by financial analysts who followed BBN, conversations with certain of my former BBN colleagues, and my own personal recollection of the events, activities and history during my years with the company, 1966 to 1997.[1]

I have included abbreviated organization charts only for BBN's Fiscal Years 1956, 1965, 1975, 1985, and 1995. These years were chosen simply because they represent an approximate mid-point in the five time periods mentioned above. However, by including only these years, I have certainly omitted the names of people and activities that undoubtedly contributed importantly to the history of BBN. To the reader and to my former colleagues I can only offer my sincere apologies.

6.2 1948–1959: Early intellectual property transfer efforts

In 1948, MIT Professors, Dr. Richard H. Bolt and Dr. Leo L. Beranek, formed a partnership to provide acoustical design services to the architects for the United Nations in New York City. They were soon joined in the venture by their former graduate students: Samuel Labate, Robert Newman and Jordan Baruch. In 1953, when the partnership was incorporated, the resulting company was named Bolt Beranek and Newman Inc. (see Figure 6.1). All five men held equal ownership interests, and Dr. Beranek served as the

Figure 6.1 Richard Bolt, Robert Newman, Leo Beranek.

Company's first President and Chief Executive Officer.

From the earliest days of its corporate existence, the founders of BBN aggressively pursued opportunities to derive additional financial return for the Company from the ideas, technology, expertise and patents that emerged from BBN's funded consulting, research and development work. (For convenience I use the term intellectual property (IP) to describe these assets.) It was BBN's standard practice to retain rights to IP developed during the course of it's work for its clients. The Company typically granted its clients a perpetual, worldwide, royalty-free right and license to use such IP in their own businesses, but BBN rarely granted them the right to sub-license that IP to others.

By the late 1950s BBN's commercialization program often resulted in licensing agreements, equity participation, and joint venture arrangements with commercial businesses which were better positioned than BBN to capitalize on the technology that BBN developed. Not surprisingly, the earliest commercialization efforts involved the licensing of IP derived principally from BBN's consulting work in the field of acoustics. Examples are included in Table 6.1.

From 1956 through 1961, BBN recorded approximately $750,000 in royalties and stock interest from its various license agreements. This represented approximately 9 percent of the $8.5 million in total revenue BBN recorded during the same period.

6.3 1960–1969: Licensing, investments, and "industrialization"

In order to repay bank loans of $325 thousand and fund a planned $500 thousand expansion of its internal product development program, in June 1961, BBN made an

Table 6.1 Early instances of intellectual property licensing

Intellectual Property	Description	Licensee
Soundstream and Aircoustat	Engine cell mufflers and air conditioning quieters	Koppers Company, Baltimore, MD
Soundsheet	Thin, plastic laminated, translucent, sound absorbing sheets that combine the functions of acoustic tile and a diffusing panel lighting system	Contrex Co., Chelsea, MA; Isora Illuminating Ceilings, Ltd., United Kingdom; Sonolite Corporation, Chelsea, MA
Audio Analgesiac System	An audio system, the "Audiac," designed to suppress pain in dental operations	Licensed initially to Bay State Electronics Corporation, Boston, MA; then to Ampex Corporation, Redwood City, CA
Space Vibration Damping	A lightweight vibration damping treatment for use in appliances	3M Co, St. Paul, MN; Lord Manufacturing
Soundshear	A series of inventions that improve the acoustic performance of sandwich panels for use in modular gypsum wall construction and panels for rigid movable partitions	Grunzweig+Hartmann, AG, Germany
Acoustic Wood Products	Acoustic wood products for the building industry	U.S. Plywood Corporation

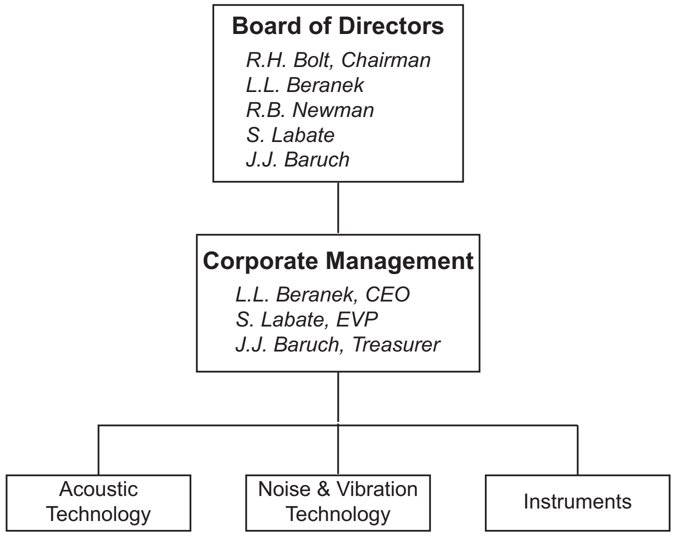


Figure 6.2 BBN Organization in 1956.

Initial Public Offering (IPO) of 160,000 of its shares at a price of $12 per share. Of the $1.9 million raised in the IPO, the company received net proceeds of $1 million and each of BBN's five founders, received $155 thousand, or a total of $755 thousand. Immediately after its IPO, BBN had a total market value of approximately $12 million. and the five BBN founders owned a total of 49 percent of the outstanding shares.

$500 thousand raised in the IPO was applied to the expansion of BBN's proprietary product development program which was concentrated in a new BBN subsidiary, Prototech Corporation established at the time of the IPO. Prototech's first president was Dr. Walter Juda, and its first Chairman was Dr. Jordan J. Baruch.

Prototech's plan was to emphasize "...inventive research in teaching machines, building materials, energy conversion, chemicals and food technology. Prototech's goal is to license its inventions under royalty agreements. When a Prototech invention seems destined to produce a marked effect in the growth of the licensee, Prototech will endeavor to secure an equity position in the licensee as part of its agreement."[2]

When it organized Prototech, BBN had no plans to manufacture products. However, in August 1962, BBN management changed it plans and organized the Honor Products Company as the Company's first manufacturing subsidiary. Honor Products manufactured and marketed "a compact, portable, pushbutton teaching machine and programs for education and training- products directly resulting from our proprietary development activities."[3]

In his letter "To Our Stockholder" contained in the BBN 1965 Annual Report, Dr. Beranek summarized BBN's policies, activities, progress, concerns, and plans related to what were then characterized as its "industrial activities". In that letter Dr. Beranek said:

"In 1961 your Company began diversifying into industrial activities in order to achieve broader returns from its investment in scientific resources. At first our industrial activities were limited mainly to royalty agreements through which we licensed BBN inventions to other firms. We later enlarged our industrial program by entering into joint endeavors with selected industrial partners, and by establishing Honor Products Company to provide specialized production capabilities. In 1964, we further extended our industrial activities by forming the Data Equipment Company.

Figure 6.3 Honor Products teaching machine.

Although the process of industrial diversification poses special problems for a professional science based firm like BBN, we are encouraged by the progress made during 1965 toward resolving these problems. We are now directly engaged in marketing five lines of industrial services and products, and we are investigating the market potential for several others.

At the same time, we are continuing to consider joint endeavors and royalty agreements to enable the Company to participate in industrial opportunities requiring resources and capabilities beyond its normal scope of operations. Such agreements will be encouraged on a selective basis."[4]

It is particularly noteworthy that Dr. Beranek and the BBN Board recognized that industrial diversification posed special problems for a professional services firm like BBN. However, notwithstanding their concerns, Dr. Beranek and the Board aggressively acted on their belief that the long term best interests of BBN's shareholders and employees would best be met by continuing to pursue the policy of industrial diversification.

In the mid to late 1960s the Company's policy of industrial diversification led to the formation of a number of new subsidiary companies and the formation of what was then an extremely significant joint venture. These are outlined in the Table 6.2.

With the formation of Prototech Corporation in 1961, BBN's efforts to capitalize on its technology accelerated dramatically. As a public company, BBN was being measured by the growth rate of its revenues and profits and BBN's employee stock option plans served as an additional pro-growth incentive to the founders, non-founder members of BBN's management, and many members of the Company's technical staff.

The first truly commercial business of BBN began with the formation of the Honor Products Company (HPC) in 1962. Originally located in St. Louis, Missouri, it was subsequently relocated to Cambridge, Massachusetts. HPC manufactured book size, electromechanical, pushbutton teaching machines that were sold through retail stores and a channel of distributors, primarily to the consumer market with special emphasis on school-age children. Its programmed course materials were embedded on specially

Table 6.2 New subsidiaries and a significant joint venture in the 1960s.

BBN Company or Joint Venture	Products/Services	Year Started	Year Discontinued	Approx. percent Owned by BBN when Discontinued	Nature of Disposition
Prototech Corporation	Developed and licensed BBN proprietary technologies in return for royalties and/or equity interests in other companies	1961	1971	20 percent	Sold to Walter Juda for 20 percent equity interest and royalties on fuel cell sales
Honor Products Company	Produced and sold teaching machines and programmed instructional materials	1962	1971	100 percent	Discontinued. In, 1971, BBN recorded $86,200 extra-ordinary loss on HPC and the Program for Advanced Study.
Muller-BBN GmbH	Provided professional acoustical consulting, research and development services in Europe	1962	1973	44 percent	Sold to Muller-BBN GmbH. BBN recorded a combined gain of $142,800 from sale of both M-B GmbH and MFE Corp. in 1973. (Gains were combined in that year for SEC reporting purposes.)
Data Equipment Company	Designed, developed and manufactured of computer input/output devices and various other computer peripherals	1963	1971	100 percent	Sold to MFE Corporation for 10 percent equity interest. Subsequently, BBN recorded a combined gain of $142,800 from sale of M-B GmbH and MFE Corp. in 1973. (Gains were combined in that year for SEC reporting purposes.)
Program for Advanced Study	Provided courses in continuing professional education, throughout the United States, for busy scientists and engineers	1964	1971	100 percent	Discontinued. In, 1971, BBN recorded $86,200 extra-ordinary loss on HPC and the Program for Advanced Study.
MEDINET	Joint venture with General Electric Company to provide real-time information services to the nation's hospital and medical community	1966	1970	0 percent	BBN received $71,500 on termination of MEDINET license

Table 6.2 *Continuation:* New subsidiaries and a significant joint venture in the 1960s.

BBN Company or Joint Venture	**Products/Services**	**Year Started**	**Year Discontinued**	**Approx. percent Owned by BBN when Discontinued**	**Nature of Disposition**
Telcomp Services	Time-shared computer services provided to scientific, engineering, and other businesses, principally in the Northeast United States	1966	1972	100 percent	Sold to On-line System, Inc. for 9 percent equity interest on which subsequent gains were $778,310
Time Sharing Limited	BBN majority owned subsidiary in the United Kingdom offered time-shared computer services provided to scientific, engineering, and other businesses in the United Kingdom	1967	1969	80 percent	Sold to the Delos International Group Inc. Recorded a gain of $779,500
Mormac-BBN Corporation	Joint venture with Moore and McCormack Co., Inc. to broadly seek commercial opportunities to apply BBN's technologies	1967	1969	50 percent (est.)	BBN acquired sole ownership in conjunction with its purchase of Wood Flong Corporation for $3.95 Million
Delos Computer Leasing	Minority ownership position in this third-party lessor of Digital Equipment Computers under an exclusive first referral agreement with DEC	1968	1976	16 percent	Sold to Automatic Data Processing Inc. for a gain of $697,600
Wood Flong Corporation	Acquired by BBN, this subsidiary manufactured matrix paper for rotary letterpress printing	1969	1973	100 percent	Sold for a loss of $352,700

designed paper rolls (much like player piano rolls) which were then inserted into the teaching machines. This course material was developed under contract to BBN's Education and Training Systems group and included programmed courses suitable for school age children as well programmed courses for industrial applications. A 1000-frame course in human relations, prepared by BBN's programming staff, and marketed by the National Foreman's Institute of Waterford, Connecticut is an example of a programmed instruction package that was designed for industrial use. [5]

In 1962, BBN and Dr. Helmut Muller formed Muller-BBN GmbH, in Munich, Germany with a modest BBN investment of $20 thousand for which it received a 45 percent share of that company's common stock. The affiliation with Muller-BBN was intended to make possible BBN's "direct participation in the industrial growth of the Common Market".[6] With Dr. Muller as its President and Dr. Beranek as its Chairman, Muller-BBN offered acoustic and noise control consulting services to clients throughout Europe, from its offices in Munich, Germany.

Prototech, also invested in a proprietary research program aimed at developing efficient fuel cells. In 1963, it was joined in this effort by the Atlantic Richfield Refining Company of Philadelphia. Together, the two companies aimed to develop a high-efficiency, compact, reliable source of electricity which in its development phase would be applied to commercial, defense, and space applications.[7] As was the case in most of Prototech's activities, a number of patent applications were filed to protect the novel and proprietary aspects of its fuel cell work.

The Data Equipment Company (DE) was acquired by BBN in early 1964 and operated as a sister division of the Honor Products Company. Based in California, the Data Equipment Company developed and manufactured a line of X-Y Plotters and other input/output and peripheral devices for computers. In addition, DE was active in the manufacture and sale of TELEPUTER consoles and controllers for time-shared computers, and the design and construction of special-purpose digital systems for data processing and for the interconnection of various peripheral equipment to digital computers.[8]

In September, 1964, BBN introduced the Program for Advanced Study (PAS). PAS was intended to serve as a program of continuing education aimed principally at scientists and engineers in the workforce. Dr. Ira Dyer who was its first Director was succeeded by Dr. Walter L. Koltun in 1968. Noted MIT Physics Professor, Dr. Phillip M. Morse, served PAS as Consultant for Academic Affairs. Courses were taught in cities around the United States, at locations convenient to the participants, often at the companies where they worked. The courses were taught by instructors affiliated with leading universities or from BBN's technical staff; all of them had extensive technical backgrounds and outstanding reputations in their respective fields. The courses, which were typically paid for by the attendees' employers, were designed to keep practicing engineers and scientists abreast of the latest technological developments in their or related technical disciplines.

In many respects, MEDINET was one of the most ambitious efforts undertaken by BBN in the 1960s. With contract sponsorship from the National Institutes of Health and the American Hospital Association, in the early 1960s BBN had designed and built one of the nation's first, time-shared, hospital information systems at the Massachusetts General Hospital. Buoyed by the encouraging prospects for the application of information technology in the health services field, BBN joined with the General Electric Company to form MEDINET in 1966. Dr. Jordan Baruch took a leave of absence from BBN to become MEDINET's first General Manager and others from BBN's technical staff were granted leaves of absence or were otherwise transferred to MEDINET to form the technical core of the new venture. "MEDINET [was] established to provide real-time information services for hospital, medical laboratories, and other elements of the medical community."[9]

Another important development during this period was BBN's creation of the TELCOMP programming language. TELCOMP was a derivative of the JOSS programming language and was designed by BBN as an interpretative language, operating in interactive mode on the first PDP-1 minicomputer manufactured by the Digital Equipment Corporation (DEC). TELCOMP was easy to learn and use, even by non-technically trained people, and it was used extensively by BBN's technical staff in the conduct of their work. In 1966, BBN began making TELCOMP available as a time-shared computer service to other companies in the greater Boston area. Demand for the service grew rapidly and additional PDP-7 mini computers were purchased from DEC and put into service at BBN's Cambridge, Massachusetts facilities. Norman Doelling served as the Vice President and Manager in charge of the new Telcomp Services Division.

In 1967, BBN was approached by Richard Evans, an entrepreneur from London, England. Mr. Evans had substantial experience in the computer field having held a number of increasingly important sales and marketing positions during his career at ICL, at that time, the largest manufacturer of computers in the United Kingdom. Mr. Evans had come to the United States to look into the possibility of starting a time-sharing service business in the United Kingdom. He had made inquiries at General Electric which was at that time selling time-sharing services based on the GE 225 computer which ran Fortran and Basic programming languages, the latter having been acquired by GE from Dartmouth College. In the end, either because he couldn't strike a favorable deal with GE or because he was impressed with the performance of the PDP-7's running TELCOMP, he and BBN entered into an arrangement whereby BBN furnished him a used PDP-1 and invested $50,000 in his new venture. In return, BBN received an 80 percent equity interest in the newly formed Time Sharing Limited with Mr. Evans serving as Managing Director. Within eleven months of its creation, Time Sharing Limited was operating profitably.

In July 1967, BBN and Moore and McCormack Company, Inc. announced the formation of a jointly owned new company called Mormac-BBN Corporation which was to concentrate its activities in the field of oceanology. General Oceanology Inc. was created as an operating subsidiary of Mormac-BBN with Dr. Ira Dyer, a Vice President of BBN, serving as its President. It was intended that General Oceanology would draw its initial staff from the ranks of BBN. At the time of its formation, the joint venture partners expressed the belief that by combining the technical expertise of BBN in the fields of oceanology and underwater acoustics, with the material resources of Moore and McCormack Company, the partners could more fully capitalize on what was then popularly viewed as the almost limitless potential of the world's oceans as a source of food, minerals, and oil.

In 1968, the minicomputer industry, led by Digital Equipment Corporation (DEC), was flourishing. However, DEC offered no rental or leasing programs for the computers they sold. Given that the cost of a minicomputer could range from a few thousand dollars to well over a million dollars, there appeared to be an opportunity to offer creative financing programs to DEC's customers and prospects, many of which were accustomed to the rental and lease programs offered by IBM for many years. It was in this context, that BBN was approached by the principals of a Boston based commercial finance company named General Discount Corporation (GDC). They proposed that BBN join with them in forming a new leasing company, dedicated exclusively to leasing and rental of DEC computers. The principals of GDC believed that for such a leasing company to be successful, it was essential that it understand the market for DEC computers and be operated by people or companies that were trusted by the leadership

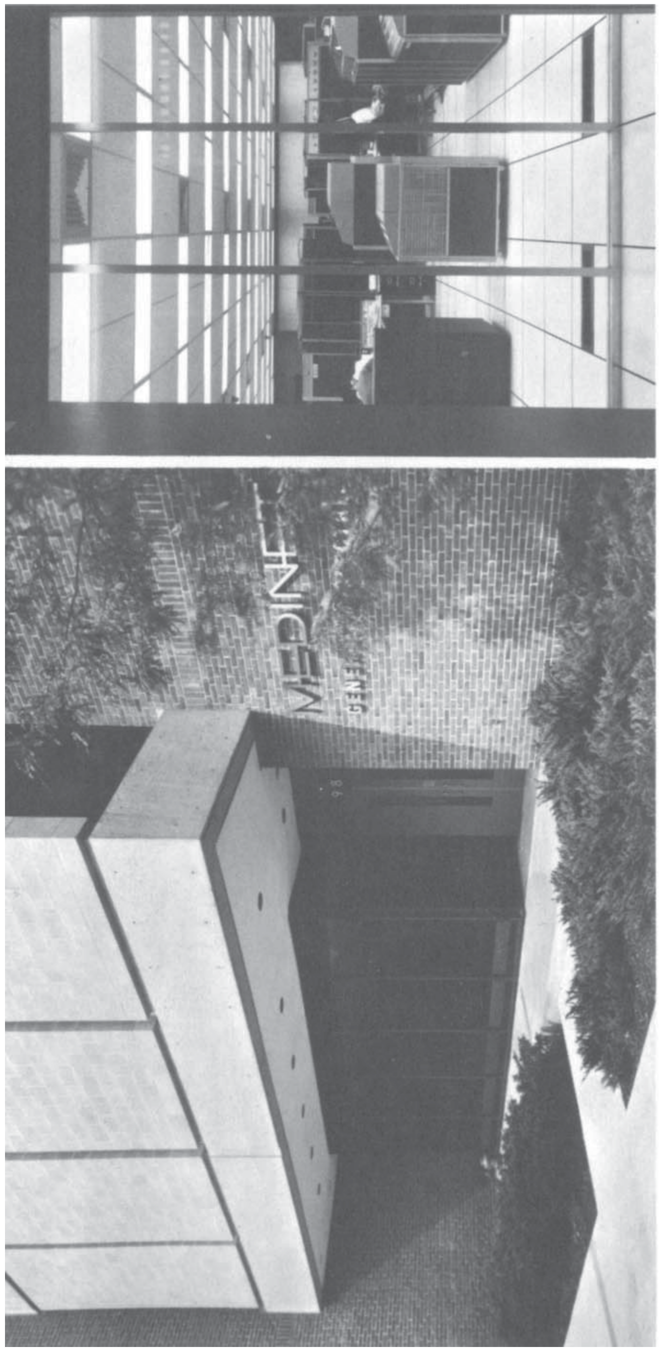


Figure 6.4 Medinet headquarters and operations center, Watertown, Massachusetts.

of DEC. Given the fact that BBN's relationship with DEC began with its purchase of the first computer that DEC had ever sold, PDP-1 Serial Number 1; and that BBN was, itself, very typical of the kinds of organizations that comprised DEC's primary market; it's not surprising that GDC believed that forming a joint venture leasing company with BBN would make sense. Delos Computer Leasing Corporation was formed in 1968, with Mr. Lawrence Seder serving as its President under a management contract with GDC and Mr. Stephen Levy serving as a Vice President under a management contract with BBN. In the same year, Delos signed an agreement with DEC wherein DEC agreed to give Delos the right of "first referral" on all potential leasing and rental opportunities generated by its worldwide sales organization.

In January 1, 1969, BBN acquired Wood Flong Corporation, a leading manufacturer of matrix paper used in rotary letter press printing from Moore and McCormack Co., Inc.. Based in Hoosick Falls, New York, Wood Flong had been in business for many years, had been consistently profitable, and BBN thought that it would provide ". . . . .an established base of earnings for our industrial businesses to build upon."[10] At the time BBN acquired Wood Flong, it also purchased Moore and McCormack's interest in Mormac-BBN which had been created two years earlier, but which had not subsequently performed according to expectations. The combined purchases cost BBN $3,950,000 in cash and short-term notes and while more than doubling the Company's profitability, and increasing its annual revenues by over 50 percent, the acquisition more than tripled BBN's Debt/Equity ratio from .25 to .82. In the years following its acquisition by BBN, Wood Flong contributed somewhat unevenly to BBN's revenues and profits.

Figure 6.5 Samuel Labate.

On July 1, 1969, Mr. Samuel Labate became the second President and Chief Executive Officer of BBN, succeeding Dr. Leo Beranek who became Chief Scientist of the Company. As has been noted, Mr. Labate, was one of BBN's founders, its first full-time employee, and the Executive Vice President and General Manager of BBN throughout the sixteen years that Dr. Beranek had served as CEO. At the same time, Mr. John Stratton, who had served as Treasurer of BBN in 1962 and Vice President, Treasurer and a Director of the Company from 1963 to 1969, was elected Executive Vice President.

6.4 1970–1979: Computer and communications subsidiaries

As BBN entered the 1970s, its "industrial activities" accounted for approximately 40 percent of its annual revenues. However, a weakening national economy began to have an adverse effect on the operating results at Wood Flong, then BBN's largest commercial

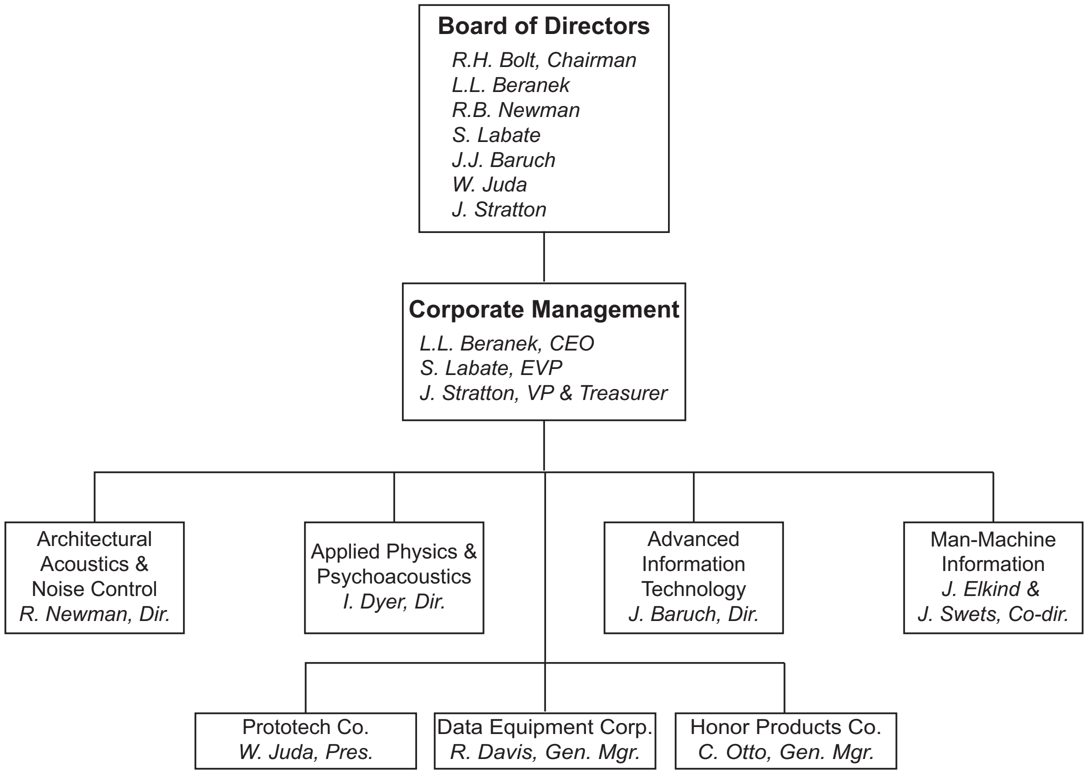


Figure 6.6 BBN organization in 1965.

business. At the same time, BBN's consulting, research and development business was growing and making important technological developments, particularly in the field of communications networking as embodied in the ARPANET project for which BBN had been the major contractor. The ARPANET project had rapidly become a major technical and operational success and BBN started to examine ways in which it might best apply its growing knowledge and experience with the packet switching technology, the foundation technology on which the ARPANET was based.

In the meantime, while BBN's Telcomp computer services business was also expanding, the Company began to search for acquisition or merger candidates that would allow Telcomp to achieve greater economies of scale in its business. Increasingly, Telcomp's large customers were directly purchasing and operating time-shared computers within their own companies and utilizing time-sharing services principally for access to the specialized applications services offered by time-sharing services companies like Telcomp. The cost of developing these specialized applications was high and BBN management believed that Telcomp would require a much larger base of business over which to amortize those costs. Discussions ensued with Graphic Controls, Inc. a Buffalo, New York based manufacturer of precision graphic paper products and the operator of a time-shared computer services business comparable in size to BBN's Telcomp Services Division.

The original intention of these discussions was to consider the merger of the two companies' respective time-sharing businesses, but those discussions were soon superceded by discussions concerning the possibility of merging the parent companies. The merger came very close to being implemented, but late in the process, SamLabate, BBN's CEO, decided to recommend to the BBN Board of Directors that it not go forward and the merger plan was dropped. Sam had come to believe that the operating style and business objectives of the management of Graphic Controls would ultimately not reconcile with the science and technology culture of BBN.

In 1972, Telcomp was sold to On-Line Systems, Inc. (OLS) , Pittsburgh, PA, in return for a 9 percent equity interest in that company and their assumption of certain Telcomp computer leases. During the course of our discussions with OLS, I met Michael LaVigna, who was at the time, OLS's Vice President of Sales. I was greatly impressed by him and a few years later when he called me to say that he was planning on leaving OLS, I suggested that he join BBN as Vice President for Business Development at BBN, a position that I had held before becoming BBN's Chief Operating Officer in 1975.

Given the earlier sale of Time Sharing Limited to the Delos International Group, Inc. (the new name of Delos Computer Leasing Corporation), the sale of Telcomp meant that BBN's interest in the computer services field by 1972 was largely represented by its ownership of minority interest in two public companies, Delos and On-Line Systems.

As can be seen from Table 6.2, during the early 1970s, a number of businesses that BBN had started in the 1960s were sold or discontinued. In general, in the 1970s, BBN Corporate Management placed a much greater emphasis on investing in commercial businesses that more closely intersected with BBN's research, development and consulting activities. There was also a significant emphasis placed on ensuring that the risk of entering a new area of commercial business was to a large degree offset by operating or capital gains from previous businesses. Thus, virtually all of the businesses that were started in the 1960s were sold or discontinued to provide capital for, and an enhance focus on, new wholly owned subsidiary businesses that were considered more promising and more closely aligned with BBN's fields of technical interest.

Table 6.3 outlines the key commercial businesses started or investments made in affiliated companies in the 1970s.

As noted above, Prototech Incorporated was created in 1961 to serve as the vehicle through which BBN's efforts at commercializing technology were to be carried out. After, Dr. Jordan Baruch was granted a leave of absence from BBN to serve as General Manager of MEDINET, Dr. Walter Juda assumed operating responsibility for Prototech as its President. Dr. Juda's interest in fuel cell technologies came to dominate the research and development agenda at Prototech and joint development ventures were entered into with the Atlantic Richfield Company and Pratt & Whitney. In 1971, BBN management decided that the investment needed to fund BBN's share of the joint development work was beyond the resources it was willing to allocate to the effort and BBN sold it's interest in Prototech to a new company with the same name, 80 percent of which was owned by Dr. Juda. BBN retained a 20 percent equity interest in the company and was to receive royalties based on the sale of any fuel cells incorporating Prototech technology.

In 1971, Dr. Juda left the BBN Board of Directors as did John E. Stratton, who had been serving as BBN's Executive Vice President and Chief Financial Officer. They were replaced by Gardner Bradlee, CEO of Cambridge Bank & Trust Company, and Dr. John Swets a BBN Senior Vice President who also served as General Manager of all BBN's consulting, research and development divisions.

In 1975, I was elected Executive Vice President and Chief Operating Officer of BBN reporting to Samuel Labate who, in that year, continued in the role of Chief Executive Officer. The following year, I was elected Chief Executive Officer to replace Sam who, as I noted earlier, had been serving as the either the General Manager or Chief Executive Officer of BBN since the Company's founding in 1948. Sam was truly a remarkable man. He was an extremely effective manager, who was quiet, modest and always gracious in

Table 6.3 Commercial businesses started in the 1970s.

BBN Subsidiary or Investment	Products/Services	Year Started	Year Discontinued	Approx. percent Owned by BBN when Discontinued	Nature of Disposition
Prototech Co., Inc.	An independent company established to market special purpose fuel cells and to continue its development of fuel-cell technology. BBN held a minority investment.	1971	N/A	20 percent	Continued as a minority equity investment, held by BBN. No royalties were received from the sale of fuel cells.
Arctech, Inc.	Professional services company specializing in cold regions technologies. BBN held a minority investment.	1971	1980	22 percent	BBN sold its 22 percent equity interest for a small gain
Hazen Research, Inc.	Provides consulting research and development services to a non-government market for mineral exploration, mining, mineral extraction and metallurgy. BBN held a minority investment.	1971	1978	2 percent	BBN sold its 2 percent equity interest for a small gain
MFE Corporation	A privately owned company that manufactures and sells analog recording instruments and digital line printers for medical and industrial markets. BBN held a minority investment.	1971	1973	10 percent	BBN recorded a gain of $142,800 from sale of M-B GmbH and MFE Corp. in 1973
Telenet Communications Corp.	Applied to the FCC for authorization to establish and operate an 18 city, United States packet-switched data communications network Initially, a BBN subsidiary.	1972	1979	24 percent	BBN sold its 24 percent equity interest in Telenet to GTE for 503,729 of its shares valued at $13.9 million

Table 6.3 ***Continuation:*** **Commercial businesses started in the 1970s.**

BBN Subsidiary or Investment	Products/Services	Year Started	Year Discontinued	Approx. percent Owned by BBN when Discontinued	Nature of Disposition
Joseph Batchelor, Ltd.	Manufacturer of low-priced mats and distributor of certain Wood Flong Products in the UK	1972	1973	100 percent	Sold as part of the Wood Flong divestiture
BBN Geomarine Services Company BBN Geoscience Corp.	Used conventional and proprietary acoustic technologies to explore the geology of the ocean floor for petroleum exploration and foundation surveys. Started as a division and became a subsidiary.	1972 1974	1977	100 percent	BBN received $1.2 million for assets, principally equipment and machinery
Autex, Inc.	Operates a real-time data system used in block trading of securities. BBN held a minority investment.	1974	1976	7 percent	Sold for a gain of $331,800
NEMEX	The New England Manufacturers Exchange provides a computerized matching service that brings buyers and qualified NE supplier together.	1975	1978	100 percent	Discontinued operations with no significant financial impact on BBN
BBN Instruments Company	Manufactured and marketed scientific instruments and transducers used to measure noise and vibration	1976	1983	100 percent	Discontinued operations with no significant financial impact on BBN
BBN Computer Corporation	Designs, develops and manufactures computers used primarily in data-communication networks	1979	1983	100 percent	Renamed BBN Communications Corporation in 1983

his dealings with people. He was also the finest mentor and dearest friend anyone one could ever ask for.

In the early 1970s, BBN made a few, relatively small minority investments in companies that operated in fields that intersected BBN's professional services activities. These included: an investment in Arctech, Incorporated, a company that provided professional services in cold regions technologies which was a field of interest to BBN's underwater acoustics division; Hazen Research Inc, a company that provided consulting research and development services to a non-government market for mineral exploration, mining, mineral extraction and metallurgy; and Autex, Inc., a company that operated a real-time data system used in block trading of securities. As the result of the sale of BBN's Data Equipment Division, BBN also held a minority interest in MFE Corporation, a private company that manufactured and sold analog recording instruments and digital line printers for medical and industrial markets. Each of these investments was sold in the 1970s and all resulted in capital gains.

In 1972, BBN organized BBN Geomarine Services Company as an operating division of BBN. The intention was to capitalize on BBN's longstanding interest and capabilities in underwater acoustics, by providing the petroleum industry with conventional and proprietary acoustic technologies to explore the geology of the ocean floor for petroleum and to provide the industry with drilling platform foundation surveys. BBN Geoscience Corporation was organized as a wholly owned subsidiary of BBN in 1974 with Mr. Ross Yeiter serving as its Chairman and Mr. Herman Sieck serving as its President. The company grew quickly and by 1975, its revenues were $7.4 million and it was operating profitably. However, its principal market, companies engaged in petroleum exploration, suffered a negative cyclical swing in 1975 and 1976 and BBN was forced to substantially cut back BBN Geoscience's operations. In 1976, BBN Geosciences experienced an operating loss, and the following year BBN sold the company for $1.2 million in cash which resulted in a capital gain of $395,000.

The New England Manufacturers Exchange was started by BBN in 1975 to provide a computerized matching service that brought buyers together with qualified New England suppliers. It was discontinued in 1978, because contract support for the service was no longer available and BBN management chose to concentrate its capital and human resources on commerical businesses it deemed more promising.

BBN Instruments Company was started as a division of BBN in 1976 with Mr. Edward Starr, a member of BBN's technical staff serving as its first General Manager. The company had its origins in the specialized instruments used in BBN professional services work in acoustics and noise control. BBN Instruments Company initially manufactured and marketed scientific instruments and transducers used to measure noise and vibration with a product line that included accelerometers and portable noise monitors. In 1979, Mr. Myron Kasok was named President of BBN Instruments Company. The following year, Mr. William Curry, joined the company as Chairman and CEO. The company, which never became a significant contributor to BBN's revenues or profitability, was sold in 1983 with minimal financial impact on BBN.

Telenet Communications Corporation was created as a wholly owned subsidiary of BBN in 1972. It grew out of BBN's pioneering work in helping to create the ARPANET in 1969. BBN's very positive early experiences as the manager of the ARPANET convinced the Company's management that packet switching technology was likely to have as profound an impact on computer-to-computer data communications as the telephone network had had on person-to-person voice communications. At that same time, the United States' regulatory environment was changing in important ways as a result of the liberalization of FCC rules that had previously inhibited competition in the telecommunications industry. Largely as a result of the Carterphone case, new telecommunications carriers were being allowed to offer communications services as "common carriers" and, importantly, they were also allowed to connect their systems to existing carriers' networks. An abbreviated form of application for carrier status was made available to potential new providers under section 214 of the Communications Act of 1934. In creating Telenet, BBN intended that it file a section 214 application to provide packet switched communications services as a common carrier, initially in eighteen U.S. cities, in much the same way that ARPA was then providing services to the U.S. research community via the ARPANET.

In 1973 and 1974 BBN invested a total of $550,000 in Telenet and as of July of 1974 it owned an 80 percent interest in the company. In view of the substantial costs that BBN had expected to incur in implementing Telenet's business plan, BBN sought additional investment partners for the venture. However, even before committing itself to the Telenet venture, BBN had first approached AT&T, then the world's dominant provider of communications services, to see if that company had any interest in building a nationwide packet-switched network. AT&T officials responded that they believed that if the technology held any potential, they were sure that their Bell Laboratories subsidiary was capable of helping AT&T implement it on its own.

BBN subsequently committed itself to the Telenet venture and created a core management team consisting of Stephen Levy as interim President, Mr. Stuart Mathison as Vice President of Business Planning and Mr. Philip Walker as Vice President of Regulatory Affairs. In 1973, Dr. Lawrence Roberts, who had played a central role in the creation of the ARPANET, left his position as ARPA's Director of the Information Processing Techniques Office and assumed the role of President and Chief Executive Officer of Telenet and I became Chairman of the Telenet Board of Directors.

BBN assembled a group of venture investors that included: Lehman Brothers, Inc., Bessemer Venture Partners, Bowne & Co Inc., and the venture arm of Time Inc.. During 1975 and 1976, these firms invested a total of $4.8 million in Telenet, and BBN invested an additional $1.4 million. These investments allowed Telenet to begin to implement its business plan. By July 1976, BBN owned 37 percent of Telenet and the company was offering its packet-switched data communications services in 43 cities across the United States.

In the March 1977, Anthony A. Barnett was elected President of Telenet and Dr. Roberts was elected Chairman of the Board. In December 1977, Telenet made an $8 million Initial Public Offering (IPO) of its shares, followed in June 1978 by a $4M secondary offering. BBN participated as an investor in both of these offerings and in June 1978 it owned a 24 percent interest in Telenet.

In the Spring of 1979, Telenet was approached by General Telephone and Electronics Corporation which was interested in acquiring the company. In June 1979, the merger of Telenet was consummated and BBN received 503,729 shares of GTE and reported a pre-tax gain of $12.2 million on its investment. The dividend on GTE shares at that time was $2.72. per share. Thus, having largely funded its investments in Telenet out of gains on earlier investments, with the consummation of the GTE merger, BBN's after-tax annual net income from dividends on its GTE shares exceeded the annual net income derived from all other BBN activities combined.

In the same year, the Company formed BBN Computer Corporation to design, develop and manufacture computers used primarily in data communications networks being built and sold by BBN. Dr. William B. Barker, who had been a member of the technical staff of BBN's Computer Systems Division and who had played a key role in designing and building the packet switches used in the ARPANET, became BBN Computer Corporation's first President. Initially, BBN Computer Corporation concentrated on the continued development and manufacture of the Pluribus multi-processor computer which had originally been developed over a period of five years within BBN's

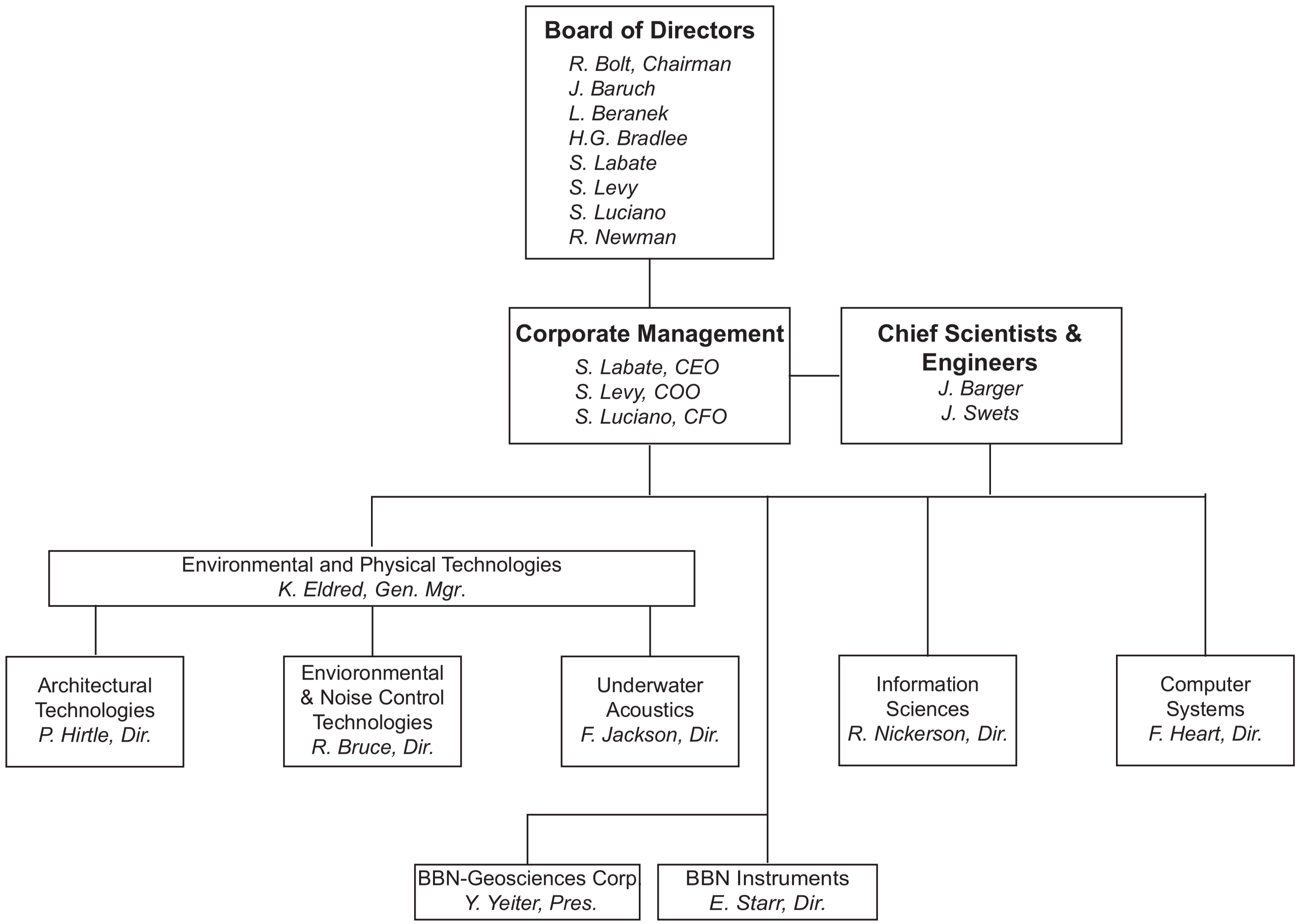


Figure 6.7 BBN organization in 1975.

Computer Systems Division. It was intended that the Pluribus be developed into a high performance packet switch for use in packet-switched data communications networks that BBN was then building for the U.S. government.

At the same time, BBN Computer Corporation also undertook further development of the "micro-programmable building block" (MBB) another BBN Computer Systems Division project. Employing what was then modern hardware technology, the MBB was designed to much more quickly execute software code that had previously been written to run on another, older, hardware platform. Initially, the MBB was used as the hardware platform for a general purpose minicomputer in data communications networks.

BBN also formed BBN Instruments Corporation as a wholly owned BBN subsidiary in 1979. Previously operated as a division of BBN, it continued to manufacture and market a line of accelerometers and portable noise monitors.

Investments in these new wholly owned BBN subsidiaries was largely made possible by BBN's sale of its interest to GTE in the merger transaction outlined above.

6.5 1980–1989: Rapid growth and the end of the cold war

As BBN entered the 1980s, management committed itself to even more aggressively using the Company's, technology, human, and capital resources to accelerate BBN's growth. It again focused on the most promising computer and communications technologies that had emerged from government sponsored consulting, research and development work, principally within its Computer Systems and Information Sciences Divisions. The strategy was largely successful and through most of the 1980s the Company achieved its growth objectives. BBN's total revenues grew from approximately $47 million in 1980 to over $300 million in 1988. Net Income grew from $2.8M in 1980 (including $1.3 million in dividends from the shares of GTE that BBN received in the GTE/Telenet merger described above), to over $18 million in 1988.

Table 6.4 lists the subsidiaries created, companies acquired and research and development limited partnerships organized during the 1980s.

BBN's first new subsidiary in 1980 was BBN Information Management Corporation which planned to develop and market proprietary software products for storing, retrieving and managing information. Two long time BBN employees, with extensive experience in software development and communications networks, Mr. David Walden and Dr. John McQuillan, were named President and Vice President respectively for the new company. The company's first product, Infomail™, was introduced less than a year after the company's formation. It was a unique electronic mail system that ran on a variety of computers and operating systems and communicated among computers using terminals and networks.[11] It should be noted that the introduction of Infomail preceded, by several years, the wide scale availability of personal computers and publicly accessible communications networks and it was therefore designed to run on timeshared computers and over private communications networks. While user response to Infomail was excellent and independent industry analysts had high praise for the product, sales outside of the technical community were below expectations. Therefore, in 1982, BBN Information Management Corporation was merged with BBN Computer Corporation where David Walden served as Chief Operating Officer and Michael LaVigna served as its Chief Executive Officer.

In March 1980 BBN Computer Corporation acquired the Lockheed Computer Corporation and its SUE and SUPERSUE minicomputer lines from Lockheed Electronics Corporation which was exiting the minicomputer business. Along with these product lines, BBN acquired a small, 100 person computer manufacturing operation located

Table 6.4 Subsidiaries, acquisitions, and R&D partnership in the 1980s.

BBN Subsidiary, Acquisition, or R&D Limited Partnership	Products/Services	Year Started	Year of Disposition	Approx. percent Owned by BBN when Discontinued	Nature of Disposition
BBN Information Management Corporation	Designed and marketed software products for storing, retrieving and communicating information	1980	1982	100 percent	Merged into BBN Computer Corporation
Lockheed Computer Corp.	BBN acquired the Lockheed SUE and Super SUE minicomputer product line and a related manufacturing facility in Hong Kong from Lockheed Electronics Corporation	1980	1993	100 percent	Became part of BBN Manufacturing Corporation until 1993 when substantially of all BBN's manufacturing was outsourced
BBN Computer Corporation name changed to: BBN Communication Corporation	Name changed to reflect the fact that its primary emphasis was in networked communications	1983	1993	100 percent	Support responsibilities for BBN Communications customers was transferred to BBN's professional services divisions, while it's Emerald ATM switch was renamed the LightStream 2010 and became the core product of LightStream Corporation in 1994
BBN Software Products Corp.	BBN subsidiary created to develop and market software products used primarily for data analysis	1984	1997	100 percent	Sold to an outside investment group for $36 million in cash
BBN RS/Expert R&D L.P.	A $3.2 million R&D Limited Partnership created to fund BBN SPC's next generation of data analysis software	1984	1987	100 percent	BBN purchased all rights to the RS/Expert technology for $9.8 million
BBN Advanced Computers Inc.	BBN subsidiary created to further develop and market products based on BBN's parallel processing technology	1986	1991	100 percent	Activities of BBN ACI were consolidated with other divisions of BBN and work on "Coral," the next generation of parallel computer was stopped

Table 6.4 *Continuation:* Subsidiaries, acquisitions, and R&D partnership in the 1980s.

BBN Subsidiary, Acquisition, or R&D Limited Partnership	**Products/Services**	**Year Started**	**Year of Disposition**	**Approx. percent Owned by BBN when Discontinued**	**Nature of Disposition**
Network Switching Systems Inc.	BBN acquired NSS for $18.2 million. NSS was in the process of developing a high capacity digital circuit switch with advanced network management capabilities	1987	1987	100 percent	Integrated with BBN Communications Corporation
BBN Integrated Switch Partners, L.P.	A $10.2 million R&D Limited Partnership created to fund BBN's development of integrated packet and circuit switching products	1987	1990	100 percent	BBN acquired all rights to the R&D L.P.'s technology for $650 thousand
Delta Graphics, Inc.	BBN acquired Delta Graphics, Inc. for $16 million. The company offered products, technology and people with skills in real-time computer graphics.	1987	1993	100 percent	Merged with BBN's SIMNET unit which was sold to Loral for $13 million ($6million in cash plus BBN's retention of $7 million in receivables)
BBN Advanced Computer R&D L.P.	A $32.3 million R&D Limited Partnership created to fund BBN's development of the next generation of parallel computers	1987	1990	100 percent	BBN acquired all rights to the R&D L.P.'s technology for $5.5 million
BBN Manufacturing Corp.	BBN subsidiary created to satisfy the manufacturing needs of all of BBN	1987	1993	100 percent	BBN ceased most internal manufacturing in favor of outsourcing manufacturing
Christian Rovsing	Acquired from Alcatel, N.V. without direct cost as part of BBN's assumption of a Delta Airlines data networking contract with Alcatel	1988	1989	1005	Closed the Christian Rovsing facilities in Denmark after completing the networking contract with Delta Airlines

in Hong Kong . At that time, the SUE and SUPERSUE minicomputers were used in BBN's Pluribus multiprocessor, a high reliability, high bandwidth switching node for packet-switched computer networks. In March 1980 BBN also introduced the C/30, a micro-programmable, medium speed packet processor. A year later the subsidiary introduced the C/70, the first computer to be designed around the C language and the popular Unix time-sharing system. In 1981 BBN Computer Corporation completed the renovation of 50,000 square feet of manufacturing and office space and established several sales offices around the United States. The company's business expanded rapidly and other new products such as the C/60 mini-computer and the BitGraph high resolution graphics display were introduced by BBN Computer Corporation in 1982.

To better reflect the primary focus of its business, BBN Computer Corporation was renamed BBN Communications Corporation in 1983 and Mr. Terrance (Terry) Fagin was named its new President. He replaced Mr. Michael P. LaVigna who had served as BBN Computer Corporation's President since 1981 and who was promoted to President, Chief Operating Officer and a Director of the parent company in 1983. In the same year, David Walden was named President of BBN Laboratories Inc., which contained the BBN's professional services divisions in which the vast majority of BBN's consulting, research and development work was carried out.

From 1980 to 1989 BBN Communications Corporation and its predecessor BBN Computer Corporation introduced a series of computer and communications products which were primarily used by government and commercial customers to build and manage private, packet- and circuit-switched communication networks. In addition to the C/30, C/60, and C/70 products, during this period BBN introduced the C/300 packet communication switch, the C/10 packet assembler and disassembler as well as the T/500 circuit switch and the T/700 circuit services manager (both products were introduced in 1987 after BBN's acquisition of Network Switching Systems Inc. which had developed them). The $18 million cash acquisition of Network Switching Systems Inc, was undertaken to expand BBN Communications' offering beyond packet-switched data networks into integrated voice and data networks. To accelerate the company's plans in these regards, BBN also organized a $10 million R&D Limited Partnership called BBN Integrated Switch Partners, Limited Partnership in the same year. In 1989, BBN Communications introduced the Netscope Software suite which was designed to facilitate network trouble shooting.

BBN Communication's networks products and services were sold to U.S. government agencies and departments, communications carriers, major international banks and credit card companies, airlines, and large industrial products and service companies around the world. A major impetus to the growth of BBN Communications came in 1982, when BBN won the Defense Data Network Contract awardedto BBN by the Defense Communications Agency in a re-procurement of the Autodin II program. In addition to this major contract, over the years BBN Communications customers included: the U.S Treasury Department, Wang Corporation, MasterCard International , MCI Telecommunications, Michigan Bell, Chemical Bank, Irving Trust, National Westminster Bank, Barclays Bank, Abbey National Bank, COMIT Bank, ENI, ISTEL, Weyerhauser, Schlumberger, Burlington Northern, KDD, System One, Japan Airlines and Delta Airlines.

BBN Communications' contract with Delta Airlines came about as a result of the Company's assumption of Delta's contract with Alcatel which itself had taken over the project as a result of that company's acquisition of the Christian Rovsing Company (Denmark). In 1988, BBN assumed the contract from Alcatel and in doing so acquired its Christian Rovsing subsidiary in Denmark. The following year, BBN Communications successfully completed the work for Delta Airlines and subsequently decided to close

Figure 6.8 A BBN Communications C/300 packet switch being checked before shipment.

the Christian Rovsing facilities in Denmark.

With the notable exception of the contract with Japan Airlines, virtually all of BBN's Communications network contracts were profitable to BBN. However, in 1989, BBN recorded a significant loss of $11M on its network contract with Japan Airlines. In the same year, BBN Communications' sales to the U.S. Defense Communications Agency dropped sharply, in part, as a result of a general decline in U.S. Defense spending following the end of the Cold War in 1989. In response to the precipitous decline in its revenues and its substantial operating losses, BBN Communications was reorganized, its headcount substantially reduced, its manufacturing plant in Scotland was closed, and its manufacturing facility in Billerica Massachusetts was consolidated into a smaller facility in Cambridge, Massachusetts.

BBN Software Products Corporation (BBN SPC) was established in 1984 with Ean M. Rankin serving as its first president. Paul A. Castleman and Channing H. Russell joined Ean at BBN SPC as Senior Vice President and Vice President of Development and Engineering, respectively. Both men had been with BBN since the 1960s and had served in a number of technical and managerial positions. The RS/1™ data analysis software that comprised the initial product offering of BBN SPC grew out of the clinical information management work undertaken by them and other members of their previous department within BBN's Systems and Technologies Division. RS/1 was "a powerful and highly integrated data analysis software package used for tasks as diverse as the analysis of laboratory data in drug research, quality control in the manufacture of semiconductors, and research and development on new fibers and textiles."[12]

To further support the development of the new subsidiary, BBN organized R/S Expert R&D Limited Partnership, a $3.2 million R&D Limited Partnership which was intended to fund the development of BBN SPC's next generation software product. (All rights to the resulting technology were subsequently purchased by BBN SPC in 1987 for $9.8 million.)

Also in 1984, for the first time since its IPO in 1961, BBN raised $16 million in capital through the sale of 707,407 shares of its common stock and in the same year listed its shares on the New York Stock Exchange.

By 1985, the operating results of BBN SPC were exceeding BBN's most optimistic projections and by 1988, its software products were in use at over 1,000 organizations around the world.

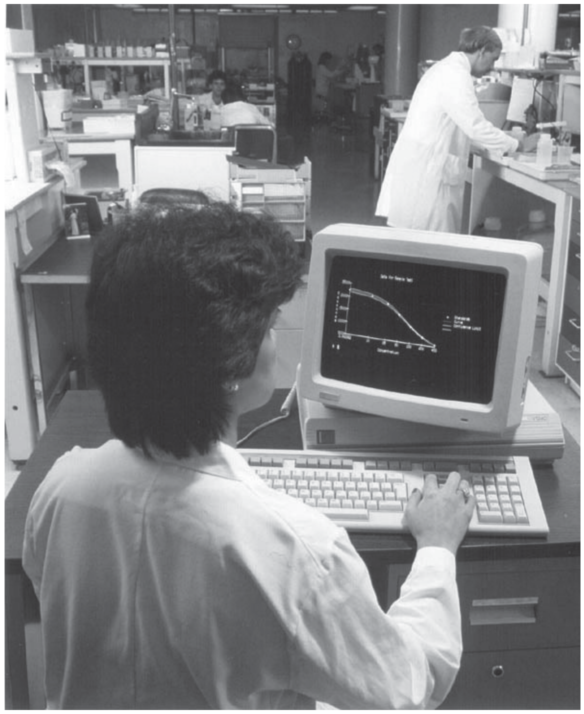


Figure 6.9. BBN Software Products' RS/1 data analysis software was used at pharmaceutical companies around the world.

In 1986, BBN formed BBN Advanced Computers Inc. (BBN ACI), to capitalize on BBN's parallel processing technology then embodied in its Butterfly™ computer, but having its origins in BBN research and development work on the Pluribus multiprocessor computer begun in 1974. Paul Castleman was named President of BBN ACI and Randall D. Rettberg and Channing Russell joined the new subsidiary as Vice President of Research and Development and Vice President of Product Development and Support, respectively.

In 1987, BBN organized BBN Advanced Computer R&D Limited Partnership, a $32 million R&D limited partnership to fund further development of BBN ACI's next generation parallel computer. In the same year, BBN raised $85 million through the sale of 25 year, 6 percent convertible, subordinated bonds.

By 1988 BBN had sold Butterfly parallel computers to DuPont, Hughes Aircraft, GTE, FMC Corp., Ford Aerospace/BDM, Martin-Marietta, General Dynamics, Boeing Computer Services, Rockwell International, RCA and a number of universities and government agencies that were investigating the use of parallel computers in their businesses.[13]

BBN acquired Seattle based Delta Graphics Inc., in 1987. "The acquisition of Delta Graphics, Inc. gave BBN products, technology, and people with skills in real-time computer graphics, one of BBN's core disciplines in the computer field."[14] BBN had become aware of Delta Graphics as a result of that company's work with BBN Laboratories on the SIMNET program which was a prototype for a new generation of interactive, networked team training simulators that were then being evaluated by the U.S. Army.

Thus, between 1980 and 1988, BBN had organized three product subsidiaries, made three acquisitions, acquired or built over 150,000 square feet of manufacturing space and over 250,000 square feet of office space, obtained over $45 million in product R&D funding through three R&D limited partnerships, raised over $16 million in equity and $85 million in 25 year subordinated debt, grew revenues from $47 million to over $300 million, and grew net income from $2.8 million to $18 million. However as the decade came to a close, BBN's business experienced a serious reversal and the Company responded with a major reorganization and substantial scaling back of its largest subsidiary, BBN Communications Corporation.

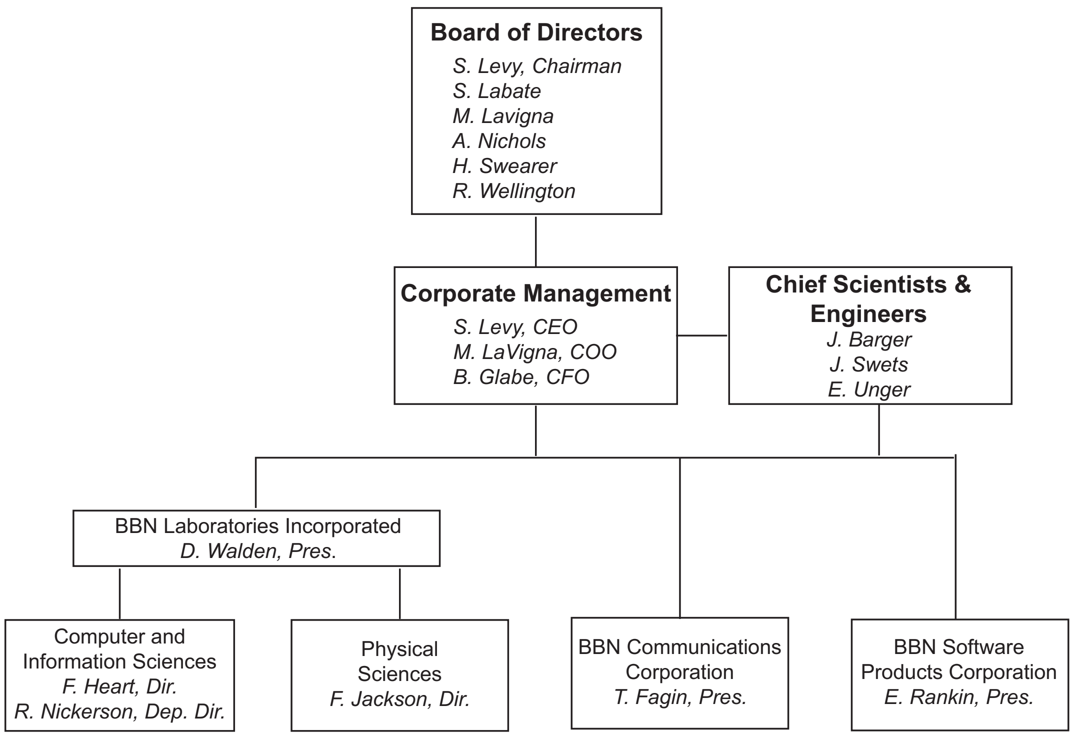


Figure 6.10 BBN organization in 1985.

6.6 1990–1997: Emergence of the Internet and acquisition by GTE

With the continued weakening U.S. economy and rapid decline in U.S. Defense spending as a result of the end of the Cold War, beginning in 1989 and continuing through 1994, BBN entered a period of declining revenues. Reductions in operating costs allowed the Company to operate at modest levels of profitability in three of those years, but from an operational standpoint, it was certainly one of the most challenging periods in BBN's history.

Table 6.5 shows the years and the nature of the products or services offered by businesses started or acquired, and the R&D limited partnerships organized by BBN in the 1990s.

As the Company entered the 1990s it faced substantially increased competition in its communications networks and software products businesses. In particular, BBN Communications faced aggressive competition in its network products business as other companies offering Internet Protocol (IP) router products started to capture an

Table 6.5 Businesses started, acquisitions and R&D partnerships in the 1990s.

BBN Subsidiary, Acquisition, or R&D Limited Partnership	Products/Services	Year Started	YearDis-continued	Nature of Disposition
LightStream Corporation	Developed and sold the Light-Stream 2010™ ATM broadband switch through a Joint venture company owned 80 percent by BBN and 20 percent by Ungermann-Bass Inc.	1994	1995	Sold to Cisco for $120 million in cash of which BBN received 83 percent
BBN Technology Services Inc. (name later changed to: BBN Internet Services Corporation, then BBN Planet, then Genuity Inc.	Initially BBN operated the New England Academic Research Network (NEARNET) under contract to MIT, Harvard University and Boston University. Accepted full ownership responsibility for NEARNET in 1994	1994	2003	After several name changes, became Genuity Inc., an independent public company, that was 10 percent owned by Verizon Communications. Acquired out of Chapter 11 for $117 million by Level 3 Communications in 2003
Bay Area Regional Research Network (BARRNET)	Acquired from Stanford University for approximately $6.5 million	1995	1995	Integrated with BBN Internet Services Corporation (then BBN Planet)
Southeastern Universities Research Association (SURAnet)	Acquired from Southeastern Universities Research Association for approximately $13 million in cash plus the assumption of $5.1 million in liabilities	1995	1995	Integrated with BBN Internet Services Corporation (then BBN Planet)
BBN Hark Systems Corporation	Designed, developed and offered speech recognition solutions to help companies improve customer service and cut costs	1995	1996	Merged back into BBN System and Technologies

increasing share of the private network communication market. While BBN Communications had actually produced the first IP based router, the T/20, and followed it with the higher performance T/200, our router product generally did not keep pace with those from companies such as Cisco and Bay Networks. These companies, and others, aggressively invested in continually improving the flexibility, speed and cost/performance characteristics of their IP router products. BBN Communications, instead, attempted to "leapfrog" to what it believed would be the next generation of communication switching products based on asynchronous transfer mode (ATM) broadband switches based on BBN's parallel processing technology.

In 1990, BBN Communications won a $32 million contract from General Telephone & Electronics (GTE) which was the prime contractor chosen by the U.S. Army to build a tactical packet-switching network for the U.S. Army. In the same year, BBN also won a $22 million contract from Wegmann & Co. GmbH to provide SIMNET technology to the West German Ministry of Defense. BBN's Software Products business continued to operate profitably in 1990, but it was another disappointing year for the Company as a whole because of lower than expected demand for BBN ACI's TC2000 computer and continued low levels of sales of BBN's communication network products to the U.S. government.

Further reductions in staffing and expenses were made and, by the fourth quarter, the Company was again operating profitably.

In 1990, BBN implemented a Total Quality Management (TQM) program in its efforts to improve it overall performance. I asked David Walden to head up BBN's TQM program on a full-time basis and Charles H. Ide was recruited from outside of BBN to replace Dave as President of BBN Systems and Technologies.[15]

In 1991 and 1992 BBN effected a turnaround in its operating results by maintaining relatively flat revenues levels while reducing operating cost. This resulted in net income of $9.5 million and $7.8 million for 1991 and 1992 respectively. During this period, the Company had pared back it operations, consolidated its BBN ACI activities into other BBN divisions, and suspended development work on a successor to the TC2000 parallel processing system, which had been code named Coral. BBN SPC continued to operate profitably, but profit margins were adversely impacted by the transition of its product line from mini computers and mainframes to workstations and personal computers. In this same period BBN Communications was transitioning its network product line from packet switching to broadband communications products with the development of the T/10 Integrated Network Access Device and the "Emerald" ATM Switch.

In the following year, 1993, BBN experienced a 10 percent decline in revenues primarily as a result of a sharp drop in sales of its systems to the U.S. Department of Defense, weak demand for the company's more mature communications and data analysis software products, and delays in developing and releasing new products. When the year began, we had projected a modest growth in revenue, thus the impact on profits from the revenue decline was much more severe than it might otherwise have been with the Company posting a loss of $32 million on revenues of $233 million. We responded with a 15 percent reduction in force of more than 300 employees, the sale of our SIMNET business to Loral for $13 million and the outsourcing of the manufacturing of our communications products. In light of the reduction in the size of the Company, we eliminated the Chief Operating Officer position and re-aligned the management of two of the operating divisions of the Company: Dr. W. B. Barker, returned from a leave of absence and was named President of BBN Communications; and Frank E. Heart was named President of the BBN Systems and Technologies replacing Charles Ide. Late in the fiscal year, we began shipment of our T/10 communications product and our Cornerstone™ desktop data analysis software product.

In early fiscal year 1994, we announced the availability of the LightStream™ 2010 ATM switch and the formation of LightStream Corporation which was 80 percent owned by BBN and 20 percent by Ungermann-Bass, a subsidiary of Tandem Computer Corporation. BBN invested $15 million and Ungermann/Bass $5M in the new company. By combining Ungermann-Bass' strengths in local area networks with BBN strengths in wide area networks, we planned to provide "total area networks" solutions more rapidly to the ATM market with the LightStream 2010 switch to be jointly marketed by both companies.

When we formed LightStream Corporation, we placed particularly heavy emphasis on ensuring that we put in place a highly experienced marketing and sales team to complement the company's strength in technology. We began this effort by recruiting a Board of Directors for LightStream that ultimately included Joseph Henson, former CEO of Prime Computer Corporation; John Shields, former Senior Vice President for Sales and Marketing at Digital Equipment Corporation; and George Conrades, former Senior Vice President and General Manager of IBM's U.S. operations. We also began a search for a highly experienced Chief Executive Officer for the company. In early 1995, Jonathan Crane joined LightStream as its CEO. Jonathan, came to LightStream from MCI Telecommunications where he played a prominent role as an Executive Vice President responsible for sales and marketing of communications services to leading businesses and other large organizations.

At about the same time, BBN created the BBN Technology Services Inc. (later named BBN Internet Services Corporation) which accepted the transfer of full responsibility for the New England Academic and Research Network (NEARNET) from MIT, Harvard, and Boston University. At the time, NEARNET had over 220 academic, research, and business subscribers and was part of the rapidly evolving national information infrastructure which became the Internet.

During this period, the computer and communication environment was evolving very rapidly as a result of the widespread deployment of personal computers, workstations and local area networks; the emergence of client/server computing architecture; the development of Mosaic and the Worldwide Web with the Internet as its communications infrastructure; the liberalization of the National Science Foundation's "acceptable commercial use" policy with regard to the NSFNET and the Internet; and the availability of substantial capital for new and established companies interested in offering new products and services in these fields. Increasingly, I came to believe that the commercial opportunities that were becoming available to BBN were enormous and that to fully capitalize on them the Company would best be served by new leadership with extensive experience in commercial markets. As a result of my work with George Conrades who, as noted above, had joined the LightStream Corporation Board of Directors in the summer of 1993, I considered him to be an ideal candidate for the position. Therefore, with the consent of BBN's Board of Directors, I recruited him to replace me and become the fourth CEO of Company. He accepted our offer, and in January 1995, he joined the Company on a full time basis as President and Chief Executive Officer and I remained as Chairman of the Board of Directors.

During his first six months as BBN's CEO, George articulated BBN's market strategy as:

> Our strategy is to provide customers with solutions to their global collaboration challenges by using all of our technical capabilities and problem solving experience in the areas of networks and distributed applications. We have a number of bases on which we can build, not the least of which is our position as a leading provider of Internet access, products, and services to more than 500 customer organizations in industry, government, education, health care, and research.[16]

As BBN entered its fiscal 1995, it had five distinct operating units: BBN Systems and Technologies Corporation; LightStream Corporation; BBN Software Products Corporation; BBN Hark Systems Corporation; and BBN Internet Service Corporation which grew out of BBN Technology Services Inc..

BBN Hark Systems Corporation was created to capitalize on BBN's Information Sciences Division's extensive technical work with speech recognition technology which had evolved to the point where powerful and practical speech recognition systems could be built to run on workstations and personal computers.

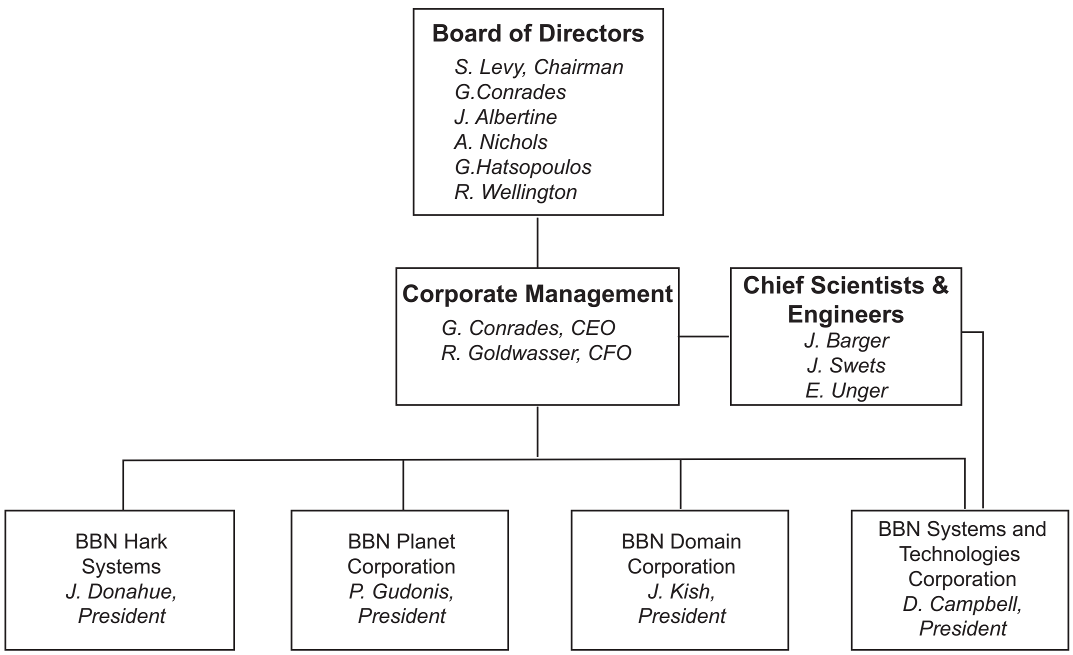


Figure 6.11 BBN organization in 1995.

In August 1994 (fiscal year 1995), BBN Internet Service Corporation acquired the Bay Area Regional Research Network (BARRNET) from Stanford University for $6.5 million in cash and stock. As with NEARNET, BARRNET had been sponsored by the National Science Foundation as one of seven NSFNET regional research networks.

During fiscal 1995, BBN also made a number of changes to the leadership of its business units including: Mr. John T. Kish, who had previously served as a Senior Vice President at Oracle Corporation, who was named President of BBN Domain Corporation (previously BBN Software Products Corporation); Mr. David N. Campbell, who had previously served as Chairman and CEO of the Computer Task Group becoming President of BBN Systems and Technologies; Ms. Julie M. Donahue, who had previously served as President and Chief Operating Officer of Voice Processing Corporation becoming President of Hark Systems Corporation; and Mr. Paul R. Gudonis, who had previously served as Vice President and General Manager-International, for EDS Corporation's Communications Industry Group, becoming President and CEO of BBN Planet (previously BBN Internet Services Corporation).

In January 1995, BBN sold its LightStream Corporation subsidiary to Cisco Systems Inc., for $120 million in cash with BBN receiving 83 percent of that amount and Ungermann-Bass and others receiving the balance. The sale of the LightStream business was intended to enable BBN to significantly increase its investment in its Internet services, software products and speech recognition businesses. Indeed shortly after the sale of LightStream, BBN announced the acquisition of SURAnet from the Southeastern Universities Research Association for approximately $13 million in cash plus the assumption of $5.1 million in liabilities. SURAnet like NEARNET and BARRNET had its origins as one of the seven NSFNET regional research networks.

As BBN ended its fiscal year 1995, it signed two important contracts to provide Internet services: one with America Online and the other with AT&T. The contract with America Online was for five years and had an initial estimated value of $55 million. It called for BBN to build and operate a portion of AOL's dial-up network in the United States. (Note: Seven years later the contract had grown to have an annual value of over $400 million per year.)

The contract with AT&T established BBN Planet as the exclusive provider of Internet access, 24 hour monitoring and managed security services to AT&T Worldnet™ MIS customers. The contract had annual options for extension for a period of up to three years, and was expected to produce approximately $120 million in revenue to BBN Planet during that term. It was intended that the contract form the basis of a strategic partnership between the two companies that would endure beyond the initial three years, however, approximately eighteen months after it was executed, each of the parties concluded that working together was not going as well as planned and entered into binding arbitration to terminate the agreement and settle the unresolved differences between them.

BBN ended fiscal year 1995 with a 10 percent increase in revenues and net income of $68.8 million which included the gain on the sale of LightStream Corporation reduced by an $18.8 million loss from operations.

As BBN ended its fiscal 1995, I announced my decision to step down as Chairman and retire after nearly 29 years as a full time employee of the Company. However, I agreed to continue to serve as a member of the BBN Board of Directors.

As BBN began its fiscal 1996, it entered into a joint venture with Andersen Consulting LLP, announcing the following:[17]

> to establish a unique, plug-in utility that offers customers a network infrastructure, 7-days-per-week, 24-hours-per-day data operations center, and a suite of reliable business applications that will enable them to conduct electronic commerce over the Internet or private intranets. BBN contributed $5 million for a 12.5 percent ownership stake in the joint venture entity; Andersen Consulting retains the remaining 87.5 percent interest. In addition, BBN ... entered into an agreement to with Andersen consulting to provide the joint venture with technical and engineering services, the value of which is expected to be approximately $4 million in fiscal 1997. The Company believes that the joint venture will generate additional demand for BBN's value-added Internet services.

The joint venture with Andersen Consulting was intended to serve as another step in BBN's efforts to focus its business on the opportunities presented by the rapid emergence of the Internet. The announcement of the joint venture with Andersen Consulting was followed in January 1996 by an announcement that Continental Cablevision was undertaking a pilot program with BBN to provide high-speed Internet access and on-line services to Continental Cablevision's home television subscribers in the Boston area.

In April 1996, BBN merged its BBN Hark Systems Corporation back into BBN Systems and Technologies and subsequently "spun-out" a portion of this business to Parlance Corporation where Jack Riley served as President. BBN retained a minority equity interest in Parlance and continued to do technical work for its new affiliate.

In June 1996, BBN completed a $54 million private placement of its common stock; and in July 1996 the Company announced that it had sold BBN Domain Corporation for $36 million.

Table 6.6 Growth rates of BBN Planet and BBN Systems and Technologies.

Business Segment	FY 1996 Revenues	FY 1995 Revenues	FY 1994 Revenues
BBN Planet	$73.0 M	$17.8 M	$7.9 M
BBN Systems and Technologies	$163.9 M	$152.6 M	$152.2 M

The effect of the private placement and the sale of BBN Domain was to further focus BBN's efforts on only two businesses: BBN Planet and BBN Systems and Technologies. By the summer of 1996 BBN had over $120 million in cash and was prepared to invest much of that in continuing the rapid growth of BBN Planet.

BBN's fiscal Year 1997 was characterized by very rapid growth of BBN Planet and continued heavy investment in that business and continued, but modest growth of its BBN Systems and Technologies business.

Table 6.6 shows the growth rate of both businesses over a period of three years:

In the winter of 1997, BBN entertained discussions with several larger companies interested in acquiring BBN, so as to acquire its BBN Planet business. In the late Spring of 1997, the BBN Board of Directors agreed to recommend to its shareholders that they accept a $29.75 share cash tender offer to be made by GTE Corporation. The effective value of the offer was $612 million plus GTE's assumption of approximately $75 million in outstanding 6 percent BBN Convertible Subordinated Debentures due in 2012, putting the total value of GTE's offer for BBN at nearly $690 million. Given that much of the market value of BBN was supported by the value of BBN Planet and that that business was likely to require substantial additional capital investment for several years, the BBN Board of Directors considered GTE's offer to be fair, reasonable and in the best interest of BBN's shareholders and its employees.

6.7 Conclusions

This brief chapter was intended to give the reader a sense of the varied technology transfer activities that took place at BBN over a period fifty years. By focusing solely on those activities, I hope that I have given the reader some insight into the nature of the process as it was carried out at BBN. I would, however, be remiss if I did not comment on the substantial effort that was also devoted to maintaining the distinctive corporate culture and drive for technical excellence that has characterized the Company throughout its history. While the push to commercialize technology was certainly driven by BBN's Board of Directors and corporate management, many members of our technical staff also considered it extremely important that their ideas and inventions be brought to market. Further, key members of the management and technical leadership teams that comprised BBN's commercial subsidiaries were often drawn from the Company's professional services divisions. Finally, management always paid great attention to providing mechanisms that would allow the "inventors" to share in any financial benefit derived from the commercial application of their work. However, so as to insulate them from the problems associated with building commercial products or services businesses, BBN's professional services divisions were organizationally (and usually physically) separated from the subsidiary companies.

When BBN decided to become a public corporation in 1961 it implicitly undertook an obligation to provide its shareholders (which included most BBN employees) the best financial returns it possibly could consistent with the high ethical standards to which it subscribed. When measured against enduring standards of corporate financial performance, technical excellence, and overall employee career satisfaction, BBN can be justifiably proud of its performance over its first fifty years in business.[18]

Acknowledgments

I sincerely thank David Walden, Ray Nickerson, and David Grier for the invaluable editorial assistance, encouragement, and helpful advice they gave me in preparing this article. I also am grateful to all my other BBN colleagues who helped me assemble and verify the information contained herein. However, for any errors of fact, only I should be held responsible.

References

1. Unless otherwise noted, all years referred to herein are BBN Fiscal Years, which ran from July 1st of the prior calendar year through June 30th of the referenced year.

2. BBN Annual Report 1961.

3. BBN Annual Report 1962.

4. BBN Annual Report 1965.

5. BBN Annual Report 1963.

6. BBN Annual Report 1962.

7. BBN Annual Report 1963.

8. Hoppin Bros. & Co. Special Study of BBN dated 8/30/67 by Arthur Okun, Research Analyst.

9. BBN Annual Report 1966.

10. BBN Annual Report 1969.

11. BBN Annual Report 1981.

12. BBN Annual Report 1984.

13. BBN Annual Report 1987.

14. BBN Annual Report 1987.

15. Shortly after Dave became leader of BBN Laboratories, it was converted from a BBN division to a BBN subsidiary and renamed BBN Systems and Technologies.

16. BBN Annual Report 1994, p. 6.

17. BBN Form 10K Fiscal Year 1996.

18. Editor's note: The next years of the business story of BBN are sketched in Chapter 22, "Epilog."

Chapter 7

Leading a Top Notch R&D Group That is Supposed to Show a Profit and Gets No Subsidy from the Bigger Corporation, While Trying to Abide by Government Contracting Rules, In the Face of Corporate and External Pressure to Take Away Researchers and Promising Projects

Frank Heart

The author notes his MIT background and transition from MIT Lincoln Laboratory to Bolt Beranek and Newman. He sketches the ARPANET project at BBN from his position as project leader, and he describes BBN's unusual mix of government-funded R&D and commercial activity, including issues and anecdotes involving government contracting, overhead rates, and employee motivation.

A history of computing at BBN can be described in several ways. For example, the chapters in this book focus primarily on many of the technical threads that BBN pursued over a 30-year period, which in some cases are still being pursued by some remaining pieces of BBN into the 21st century. One of those threads, computer networking, was especially important to the world. It was a transforming event causing major growth and changes at BBN, and it eventually led to the 1997 sale and breakup of BBN. But another, orthogonal way of considering the history of computing at BBN is to discuss the BBN research environment, the ways in which it differed from other companies and from universities, and how BBN interacted with its primary client, the federal government.

Because my career intersected both the computer network technical thread and the management of a sizable segment of the BBN research environment, and because my pre-BBN experience had some impact on both, this chapter will first mention my pre-BBN years, and then discuss various aspects of the BBN research environment that impacted most of the technical threads that other chapters discuss.

7.1 MIT and Lincoln Laboratory

Unlike the careers of many more mobile people, my career consisted of only two jobs: Massachusetts Institute of Technology (MIT), for about 15 years, and BBN for nearly 30 years. As a junior at MIT in 1950, I discovered that such a thing as a computer existed: Whirlwind,[1] an early electronic computer, had just begun to operate at MIT. At that time an Englishman, Gordon Welchman, was teaching what was the first programming course at MIT[2] (and I believe was probably the first programming course in the United States). I joined the Whirlwind staff as a research assistant while obtaining a master's degree at MIT. At that time, Whirlwind used vacuum tube electronics and had a grand total memory of 32 registers into which instructions and data could be entered with manual toggle switches. (This soon changed to 256 registers of electrostatic tube storage, then to 512 registers of the newly invented core memory storage.) Although originally supported by the U.S. Navy, Whirlwind began receiving support from the U.S. Air Force as part of an effort to build an air defense system. The group working at Whirlwind on air defense, including me, shortly became part of MIT's Lincoln Laboratory, which moved to a new home in Lexington, Massachusetts, where we worked on the Semi-Automatic Ground Environment (SAGE) air defense system.[3]

Sometimes ideas learned early in a career make a large impact on later activities. Because electronics in general, and computers in particular, were unreliable in those early days, the person managing the Whirlwind project, Jay Forrester, was very concerned with reliability issues, and highly specialized techniques were used to attempt to ameliorate the reliability flaws in Whirlwind and the follow-on electronics of the SAGE system. I learned this lesson well, and many years later, in managing the building of the ARPANET, I believe that my emphasis on reliability made the difference between success and failure of the ARPANET.

Early air defense experiments with Whirlwind involved the Cape Cod System, wherein radars on Massachusetts' Cape Cod were connected by phone lines to the Whirlwind computer. The computer had to deal with the radar data in real time, that is, the computer had to accept the data at the phone line rates and deal with each radar scan before the next one came along.

This kind of computer use was unusual at the time, but at Lincoln Laboratory the group of people working with me became unusually expert at the real-time use of computers. At Lincoln, in a long series of projects, computers were connected to various radar antenna systems, radio antenna systems, sensors at underground seismic arrays, and sensors at underwater acoustic arrays. Each such project required a detailed understanding of the computer timing relative to the time sequence of data arriving from the various sensor systems. This experience at Lincoln — tying computers to phone lines, and constructing hardware and computer programs that involved the timing constraints of such data handling — was a crucial attribute of my group at BBN that years later bid on and won the ARPANET contract.

In mid-1965, the then-director of Lincoln Laboratory, Carl Overhage, managed a multiorganization working conference, called Intrex,[4] at Woods Hole on Cape Cod, and I was asked to participate. Aside from my interest in the specific purpose of the conference, which concerned the use of computers in libraries, the conference had two other unrelated impacts on my life. First, my third child was born while my family and I were staying at Woods Hole, necessitating a harrowing but in-time, floored-accelerator car ride back to the hospital in Boston. And second, I met and became friends with Danny Bobrow, an AI researcher who worked at BBN. A year later, when BBN was seeking a manager for a National Institutes of Health-funded project to use computers in hospitals, Danny apparently remembered me, and BBN's Dick Bolt embarked on a project to extract me from Lincoln Laboratory. I was happy at Lincoln, and had a strong group of people working for me on various computer systems, and I was rather conservative, so this extraction was not so easy, but it ultimately succeeded, and in December 1966 I joined BBN.

7.2 BBN in late 1966

When I arrived, BBN had been in business for more than 18 years, had more than 400 employees, and had a sales volume of over $7,000,000. Quoting from the 1967 annual report of the company:

> We combine activities of two types. Through consulting, research, and development in fields of applied physics, acoustics and noise control, information science and technology, applied chemistry, and education and training, we derive new knowledge and solve specific problems for our clients and sponsors. In these same fields, we meet more general needs by marketing industrial services and products[5]

The consulting, research, and development activities represented the core business of the company when I joined it, and remained the company's core business over the years of my employment. I will primarily focus on that area and discuss some of the company's service and product activities as they relate to the core R&D business.

Many U.S. corporations have R&D components, and in most cases those components represent an expense (or investment) of the corporation, with the money coming from other corporate activities. The pharmaceutical companies are obvious examples, and some well-known examples in the computer field include the IBM research laboratories, the Bell Laboratories, and the Xerox Palo Alto Research Center. In each case, the company's product activities earned the money, and the R&D groups spent the money — with the hope, no doubt, that new moneymaking products might arise from the research and development. This common model had no relation to what prevailed at BBN.

At BBN, the company hired bright people and then expected them to find support for their consulting, research, or development activities. The modest profits from these opportunities were then used by the company to explore industrial service and product activities, in hopes that the industrial and product activities would someday make money. Somewhat amazingly, this unusual model worked well for decades. These consulting and R&D activities varied in size from a one-person consulting job in architectural acoustics to large, multiperson, system development activities.

BBN hired me to supervise several groups, including one such multiperson development activity that had been running for more than four years — the Hospital Computer Project — an activity supported by the National Institutes of Health (NIH), and operated jointly with Massachusetts General Hospital. This interesting early medical information project, as well as several follow-on medical information projects pursued by my group, is discussed in some detail in another chapter (Chapter 12). Unfortunately, the Hospital Computer Project with Mass General was in some trouble when I arrived, and my arrival wasn't enough to avoid an eventual termination of BBN's role in the project. Mass General, of course, continued evolving the use of computer systems over the succeeding years.

7.3 BBN R&D environment

The R&D environment was influenced by many factors, including clients, rate structure, and timekeeping practices, among other things.

Client base

The consulting, research, and development activities were supported by a diverse client base. The architectural acoustics and noise control activities primarily were supported by commercial clients, although both federal and state governments used these services at times. The work we did in physical sciences and in information sciences was primarily for various federal agencies. Because this history is concerned with computer research and development, I will primarily discuss the client base for the groups doing some form of information processing. It's perhaps obvious by now that BBN was a complex company, spanning many different disciplines and businesses: a confusing entity from management, financial, personnel, reward structure, risk, and legal viewpoints.

The client base for the groups doing information processing research and development was heavily weighted to the federal government. Clients included primarily various agencies in the Defense Department, the National Science Foundation (NSF), the NIH, and occasionally other federal agencies such as the Federal Aviation Administration, Internal Revenue Service (IRS), and so forth. The easiest client to deal with was the Defense Department because it had a long history of dealing with profit-seeking firms, had well-defined contracting approaches, was willing to pay the rather high BBN rates, and was interested in many of the technical areas favored by the BBN staff. Even within the Defense Department, there was considerable variation in client behavior. Some defense agencies were staffed by extremely bright technical people and working relations were unusually collegial. Other agencies were more bureaucratic, had a less capable staff, and wanted a more arms-length relationship with BBN. The groups doing information processing dealt especially well with the Advanced Research Projects Agency (ARPA, or sometimes DARPA, for Defense Advanced Research Projects Agency).

BBN had more difficulty dealing with federal agencies that usually dealt with universities and nonprofit institutions. Those agencies, especially the NSF, but also the NIH, were not really comfortable dealing with profit-seeking organizations. There was a general view that profit-making organizations normally *gave* money to universities, rather than competing with universities for federal funds. Further, the "study sections" used by the NSF and the NIH to evaluate proposed activities were packed with academics, and representatives of the profit-seeking sector were few and far between. (The fact that BBN was profit-seeking but actually did not exhibit great profit growth did not impress those government agencies.) So, even when some technical part of the NSF or NIH was interested in doing business with BBN, arguments would ensue about whether the grant or contract would allow any fee or, for that matter, would allow the normal BBN overhead rates, which were much higher than those at academic institutions. It was a testament to the quality of BBN researchers in the information sciences and in education that, despite these difficulties, we did get NSF and NIH contracts and grants, often with reduced fees and overhead, but occasionally with normal rates.

Some clients represented unusually disappointing outcomes. Because of complex client management structures and internecine war-fare between various parts of the client organization, BBN was unable to help some clients even when, technically, we were in a position to do so. A particularly egregious example was the IRS. BBN received multiyear funding from a research component of the IRS, and we developed technology that (in my view) might have improved tax collection by substantial amounts. However, the actual technology deployment decisions at the IRS were in the hands of various IRS feuding components, none of which was the component funding BBN. So after a few years, the effort simply died. We had been so sure that we could help the IRS that we actually tried lobbying to get the attention of IRS top management, but in vain.

Thus, there was a premium on trying to get involved with individuals in an influential position with a prospective client. Some of BBN's R&D groups were quite clever at connecting to a person or subdivision of a client that was in a strong position with that client, but this wasn't always possible.

There were some client interactions where BBN played a useful role but never received adequate credit. Sometimes the client didn't want to give credit to a contractor; sometimes BBN was part of a group of contractors or served as a subcontractor to a prime contractor who wanted the bulk of any available glory; and sometimes issues of security classification precluded good publicity. Then, too, sometimes BBN initiated an important project that was continued by others, and BBN's role was lost in the overall project story.

I was personally involved in securing one such project where BBN's role has been forgotten. Today, Genbank is a nationally funded central component of the worldwide efforts in understanding the human genome and all the related biological research surrounding such efforts. Although almost nobody knows this, BBN was instrumental in Genbank's initial development. BBN had contracts with the NIH, which, among other things, supported a small group of BBN computer scientists who also understood molecular biology. Another group of scientists, at the Los Alamos National Laboratory, was the most knowledgeable group in the country concerning the technology required for a database project such as Genbank. When the NIH put out a request for proposals for the initial Genbank contract, there was a curious problem — Los Alamos, as a creature of the Department of Energy, could not bid competitively for such an NIH contract but could accept a sole-source subcontract from somebody. BBN was most interested in winning the Genbank contract, and after rather convoluted negotiations, BBN bid as prime contractor with Los Alamos as a subcontractor. This meant money would flow from the NIH to BBN, back to the Department of Energy, and on to Los Alamos — very strange. This contract lasted several years, and then was recompeted by the NIH. For a variety of reasons, Los Alamos was not happy with BBN's behavior as prime contractor and chose to team with another prime contractor. Nonetheless, we really did get Genbank off to a good start.

An interesting aspect of the client base relates to how the various BBN R&D groups secured new contracts for new activities. Many federal agencies, and all those with which BBN was involved (with certain classified exceptions), were required by law to announce new work in the *Commerce Business Daily (CBD)* so that anyone who felt qualified could respond to try to obtain the work. Sometimes BBN actually obtained some new work by noticing such an announcement and responding. However, in general, if one found out about new work possibilities by that route it was usually already far too late. BBN scientists developed relationships with client organizations, which allowed staying in touch with the generalized future plans of such clients. BBN could think about things that such a client *might* want or that BBN thought the client *should* want, and thus be much better prepared to respond when the client finally decided what it wanted and announced such plans in the *CBD*. This was especially useful when a client didn't know precisely what it wanted and stated its desires in general terms, asking responding organizations, in effect, to define both the problem and the solutions. Once again, some R&D groups at BBN were better at this client prediction process than others, and BBN placed a considerable premium on maintaining relations with clients that allowed BBN to be prepared for the next client need.

It is worth noting one aspect of contracting for research and development with government agencies in which the Congress, in a well-intentioned search for "fairness," managed (in my view) to shoot itself in the foot. In my early years at BBN, most federal agencies contracted for work by a combination of sole-source contracting and competitive contracting. Then Congress became unhappy over certain egregious behavior of some federal agencies in the use of sole-source contracts and mandated that almost all contracts must be let by competitive bidding, with exceptions requiring high-level approval. At first blush, one might think this was a good step, but in fact it placed a great burden on the federal agencies. The mandate led to awards to wholly inappropriate contractors that chose to submit unrealistically low bids, and it led to awards to contractors whose past performances were abysmal but whose current bid could not be dismissed on the basis of that performance. It also meant that the process of bidding, for a contractor like BBN, was much more expensive, lengthy, and fraught with risk of someone incompetent "buying in." Although BBN, in a fair competition, could usually do well, not all competitive bidding situations were (in my view) fair.

Contracting and the BBN rate structure

Although contract issues and financial rate structure may be viewed as dull, overshadowed by more fascinating discussions of computer technology, such issues had a first-order impact on how BBN conducted research. Contract issues also affected the availability of funds for new technical thrusts, on staffing, and on the competition with other organizations for contracts. Most of the computer research and development at BBN was conducted with the federal government under cost-plus-fixed-fee (CPFF) contracts, wherein BBN bid a total expected price for some job, but provided the government with information about how that price was built up in actual costs plus a fixed dollar amount of fee (profit). Then, in the course of the work, the government would be charged what the job actually cost, plus incremental fractions of the negotiated fixed fee. If the job was done for less than the original expected total, BBN would still get the proposed fixed fee, but costs would be less. If the job was overrun, BBN would ask the government if it wanted to continue absorbing the excess costs or if the job should be discontinued, but BBN would not get more than the original negotiated fixed fee. BBN always strove not to overrun jobs because it would annoy the client, even if the client agreed to additional costs to finish the job. Sometimes this worry about client annoyance would lead BBN to "eat" the fee in order to finish the job without asking for overrun funding — but this of course annoyed the BBN management. BBN did do some fixed-price contract work where the total price was set at the outset, whether the job took less or more than that amount to finish, but this was the exception, and was somewhat dangerous for research and development, because the very nature of advanced research is the uncertainty of how hard it may be to do the job.

The rate structure led to the total price for job labor as the product of four factors:

- Direct Labor (DL) — the actual costs, salary, and benefits of the people who would work on the job.
- Overhead (OH) — all the costs that could be sensibly allocated to support the people working on the job, especially the down-time of technical people and time spent by technical people on marketing or proposal writing, but also on such items as space costs; contract, finance, and administrative support staff; communications, computer usage, and many other allocable costs.
- General and Administrative (G&A) — remaining costs that could not conveniently be allocated to specific scientists or specific jobs, such as corporate management, finance and legal staffs, security, and so on.
- Profit.

This rate structure, mirroring actual costs, resulted in a total price for a scientist's labor of between 2.5 and 3.0 times the actual salary of the scientist (the "multiplier"), a figure considerably higher than comparable prices from a nonprofit institution such as a university. This expensive nature of BBN labor was a problem in competitive situations, as well as a political problem in trying to obtain contracts or grants from federal agencies more used to dealing with universities and often unwilling to pay profit at all. More generally, it meant that BBN had better be exceedingly good at what it did in order to command such high prices.

Finally, the rate structure also caused internal problems within BBN, wherein people in various departments, faced with customer demand for lower prices, resented many of the costs imposed on their labor by the central corporation. As in any case when assessment of costs is not within one's control, one tends to find the assessment too high or unnecessary. This problem was especially severe in relation to satellite offices outside of BBN's Cambridge, Massachusetts, headquarters; in remote offices, employees resented the high rates and constantly lobbied for lower rates in order to improve their competitive position, on the basis that they did not benefit from some of the services provided in Cambridge for which they essentially were required to contribute (such as a cafeteria in the Cambridge facility).

Federal government contracting procedures contained one key benefit for organizations such as BBN; specifically, they allowed contractors to include in the rate structure certain costs for independent research and development (IR&D). Of course, including IR&D costs raised the overall multiplier and impacted BBN's competitive position, but the availability of IR&D funds allowed departments to investigate new ideas, new fields, and new problem approaches to better position the company to obtain new contracts or grants. Each year, many ideas arose for possible IR&D projects, and sensible selection of such ideas for support with limited funds was important to future success and growth.

Staffing and chargeability

Even as the company grew, BBN assumed that individuals and departments would find their own contract or grant support and thus earn enough to pay for the department expenses or, even better, enough to also help support other individuals or departments. This created a constant pressure on individuals and departments to find work. The mathematics of the rate structure led to a necessity for the departments to be, roughly, at least 70 percent "chargeable": that is, to have enough contract jobs that, on average, every individual in a department could charge 70 percent of his or her time to a specific funded job. The remainder could be charged to overhead or other non-funded activities to seek new work, write proposals, do internally funded research, and so on. This chargeability was closely monitored by the departments and by the division and corporate management. The ease of meeting this goal varied widely between departments. For example, a department with one or more large, long-term jobs might be nearly 100 percent chargeable, because everybody was working all the time on one of the large jobs. Conversely, a department with a surfeit of small jobs, interspersed with periods of looking for more work, might always struggle to reach the 70 percent goal. Thus the departments with lots of work would, in effect, allow some subsidization of valuable groups that temporarily could not meet chargeability goals, as long as the BBN division (group of departments) on the whole met the chargeability goals.

Groups that were doing well could expand, spend more freely on acquiring new work — and generally had more smiling managers — while groups that struggled to meet these goals might have to contract, or reduce some staff to part time, and generally had managers who were frowning more of the time.

Clearly, there was a premium on attracting and keeping staff who could attract work, manage the work's performance, and keep the clients happy. There was some variability in the care with which departments screened potential staff, but on the whole BBN had an exceptionally talented staff. This was especially crucial for BBN, because most of the marketing for new work in the R&D departments was done by the senior technical staff, not by a marketing department. Often, individual staff members had long-term good relations with some segment of the client community, especially at ARPA, the National Security Agency (NSA), or the NIH, and were able to keep contract funds flowing without interruption for years.

Timekeeping complexities

Closely related to the chargeability issue was the mechanism BBN used to track the time spent on various different activities. The accounting rules required that each individual fill out a time sheet each day, recording the time spent on each funded job, on each kind of overhead activity — such as proposal writing and marketing — or on various other activities such as sick leave or vacation. Filling out time sheets was viewed as a nuisance, uniformly disliked, but it also caused the technical staff more serious annoyances that were really some blend of moral and legal behavior.

The most typical difficulty was how to treat work done in excess of the normal workday, either at the office or at home. Most scientists do not forget their work when the clock strikes 5:00 p.m., and there are strong incentives to finish jobs in a professional manner, on time, on budget, and to the client's satisfaction. So if a job needed extra work, many scientists would put in the extra hours to get that work done. But how to charge that time? If the hours were charged to the job, it might well overrun the funds allocated to the job, eating up the already meager profit margin. If the time were not charged, it might be considered a violation of the time-keeping policy, with various associated risks. Similarly, when writing proposals or marketing, if the staff worked overtime to produce a great proposal, should the overtime hours be charged to overhead (marketing) or not? If charged, it would reduce the individual's and the group's chargeability, leaving less for other overhead activities and potentially causing questions about staffing levels.

Still another serious question related to scientists who were simultaneously working on both government and commercial jobs. The government was concerned about time charged to the government but which was actually spent on activities related to seeking or performing commercial business. These issues required a small but constant level of attention on the part of the group managers and the individuals, and the managers needed to regularly remind individuals of the importance of following the government rules.

Another difficulty related to "hacking," or working on technically interesting activities that were not really related to company business. Scientists often found interesting issues in their personal lives that were amenable to computer support — such as producing cave maps, providing bookkeeping support to a religious organization, or working on a bridge game. Because often the best scientists were those who had such outside interests, the company tried not to discourage limited amounts of such hacking, but it required careful accounting for computer usage and management attention to ensure that government accounting rules were followed.

Reward structure

With a staff of highly talented individuals, all holding strong views about their value to the company, it was a challenge to deal properly with compensation issues. BBN did not publish salaries, but the environment made it likely that people could and did have a good idea of other individuals' compensation. This was partly because, in government contracting, anybody managing a CPFF contract would have access to the cost buildup of the price to the government, including the costs for each individual working on that contract. Consequently, BBN had to carefully ensure that compensation decisions were reasonably defensible — maybe not perfectly defensible, but not so unreasonable as to infuriate people.

For salary determination, people were grouped into levels based on education, years of experience, management role if any, hiring salary level, and other less-precise factors. Those levels then translated into salary ranges, and an attempt was made to give raises based primarily on performance but also on the relative position of individuals within the level-determined salary range. Personally, I also used an approach that required managers to provide lists of department members in order of total (long-term and short-term) value to the company. When managers had difficulty with such ordering decisions, I suggested that they imagine two people coming in to resign and guess which one whose mind they would try harder to change. Unfortunately, such approaches, however creative, did not develop a sufficiently wide gap between the best performers and the rest of the individuals.

This difficulty was especially galling at the lowest levels — top performers among the secretarial and other supporting staff — where the personnel department and corporate management felt that BBN had to compete in a regional labor market and did not want salaries to be out of line in that marketplace. However, the limited salary gap between the best performers and the rest was also a serious problem at the high end, where many BBN staff were good enough to be receiving frequent job offers from elsewhere. Although BBN's location, ambiance, enlightened management, and cutting-edge research and development went a long way toward keeping people from jumping ship, those with growing families did notice the compensation issue as well.

Two other tools were used to ameliorate this problem at the high end. For a few of the very most productive people, BBN awarded stock options. In the 1970s and 1980s, such options, while seldom overly generous, did provide a few people with an incentive to stick around. A more useful tool, during the years that the company was doing well financially, was a bonus plan. Jim Barger, another senior BBN manager, and I invented a plan that allowed a wide disparity in bonus allocation, specifically to address the limited variation in salary. In a few of the most profitable years, for a few of the best people, this plan generated bonus amounts as high as 30 percent of salary. Unfortunately, the lifetime of this approach was limited, and as the company experienced less-lucrative years, compensation always tended back toward modest variability.

Classification

Because working on classified projects required extra care and effort, and because some BBN staff preferred to avoid military projects, BBN was fortunate that most of the contract work for the U.S. government and even most of the work for the Defense Department was unclassified. However, because some of the work in many departments was classified, most of the management and senior technical staff held security clearances. The underwater acoustics work was frequently classified, as was computer and network R&D for certain agencies. In fact, the classification question occasionally seemed more determined by who was the sponsor rather than by relation to the particular task of the moment. BBN also took a small amount of highly classified work that required special procedures, but in most cases clearances for such work were limited to just the few people actively working on the job; even the management was not cleared into the program. BBN was always extremely careful to follow all of the rules concerning classified work, but it was always a considerable nuisance, and we were sometimes able to convince the sponsor to limit the technical areas of an overall contract that required classification.

Affirmative action

BBN, along with the rest of the United States, was subject to various pressures from the federal and state governments to deal appropriately with minorities and female employees. There were really two sets of issues that required attention from the staff and the management. The first was to ensure equitable treatment of existing employees with regard to job assignments, reward structure, titles, workplace rules, and inter-employee behavior. The second was to actively seek new employees in such a way as to increase the percentages of females and minorities in the BBN work force.

The first issue was reasonably easy to deal with. Because BBN's R&D success depended on the quality of the scientists, BBN was, for the most part, a meritocracy, and minority and female scientists who were good at the job were treated as well as, or often even better than, other people. The only difficult issue, with regard to existing employees, related to individuals who were doing badly or having trouble. We had to exercise special care in such cases, in order to avoid any hint of bias. BBN sometimes "carried" such a person for longer or tried harder to reassign and keep such employees.

The second issue, to actively find more qualified minority or female scientists, was much harder. BBN tried recruiting at minority colleges and tried hard to attract qualified female scientists, but it was difficult to meet goals set by the government and, in turn, by BBN's internal personnel group. It was simply an availability issue — not enough qualified minorities or females could be lured to the door. Not enough minorities and females were seeking technical careers or going to good technical colleges. Again, because BBN depended on the scientists for cutting-edge R&D, we needed very good people and could only hire the ones we could find.

7.4 ARPANET and its impact on BBN

In 1968, my group bid and won the contract for constructing the ARPANET, and we dealt with a smart group at ARPA in the many years during which that project grew from its inception to the genesis of the worldwide Internet. Although the network activities are discussed more fully in another chapter (Chapter 17) the ARPANET was sufficiently important to my years at BBN for me to discuss it as well.

In 1967 and 1968, ARPA began to consider the idea of computer networking. In a story that has been told in many other places,[7] Bob Taylor, head of the ARPA Information Processing Techniques Office convinced his management to fund initial work; Bob attracted Larry Roberts, a computer scientist at MIT's Lincoln Laboratory, to ARPA to lead the program; and Larry began considering how to proceed. By early 1968, ARPA had decided to proceed with a procurement of such a computer network, and Larry began talking to various potential contractors about participating in such a venture. I believe that Larry had initially hoped to interest AT&T or IBM in such a venture but, for various reasons, these companies did not wish to participate.

Some people at BBN, including Bob Kahn, a theoretician working in the Information Sciences Division, already had been involved with ARPA during 1967 and 1968 in considering some technical aspects of the network idea. I first heard about the plans at the Spring Joint Computer Conference in 1968 in Atlantic City, New Jersey. I had known Larry at Lincoln Laboratory, and when we met on the boardwalk in Atlantic City, he indicated that ARPA was considering the procurement of a computer network, that he was talking to various potential contractors, and that perhaps BBN might want to consider some involvement. Larry knew that I, and the group of people who had followed me to BBN from Lincoln, were expert in the connection of real-time systems to computers and that such expertise might be applicable to the computer network arena.

BBN took this potential procurement seriously, and began thinking a little about the technical issues even before the actual request for proposals arrived. When it did arrive, I led a team of people (including Will Crowther, Bob Kahn, Severo Ornstein, and Dave Walden) in a crash effort to write the winning proposal, and on 1 January 1969, BBN was awarded the ARPANET contract to build a four-node network, with the possibility of expansion to additional nodes. Thus began the Internet.[6]

Figure 7.1. The author with one of the original ARPANET packet switches. (Photo from the author's collection.)

The ARPANET contract led to a stream of contract R&D for more than two decades, some directly from ARPA as well as some from other Defense Department agencies, the NSF, the NSA, commercial organizations such as large banks and large airlines, and from foreign governments. These R&D contracts involved

- expanding the ARPANET;
- building satellite- and radio-based networks for ARPA;
- building other independent networks for federal and commercial clients;
- designing, manufacturing, and delivering new kinds of computer systems for these other networks;
- developing, delivering, and operating monitoring systems for network management; and
- many other related activities. For the years covered by this history project, these activities were successful, exciting, and profitable.

At the time of winning the ARPANET contract, BBN had essentially no manufacturing capability and relied on Honeywell to construct the specialized I/O hardware needed to transform a standard Honeywell 516 computer into an Interface Message Processor

(IMP). In the next few years, based on sales of networks, BBN established a modest-size factory and began producing the many kinds of specialized hardware devices required for the many networks being built.

As the ARPANET's success became more widely known, there was a surge of interest in the project both within the United States and from groups in many other countries. BBN became a bit of a tourist stop for many groups from overseas; although in some ways we liked the attention and were happy to spread the word, at one point the traffic became so demanding that we considered charging for such sightseeing visits by outside groups.

As the ARPANET grew, many organizations around the country that were not ARPA contractors (especially universities) did not have access to an ARPANET connection. Consequently, pressure began to build on the NSF to enter the game in order to provide such network service to the ARPA-have-nots. This led to the development around the country of so-called NSF regional networks, built and operated primarily by various university consortiums. After watching this series of developments for a while, it became clear that New England needed such a regional network, and BBN encouraged MIT, Harvard University, and Boston University to jointly sponsor such a network, called Nearnet. The universities needed a network operator to actually build and run the new network, and BBN won the contract to build and operate Nearnet.

This stream of R&D network success led to a series of attempts to augment the network R&D activities with serious commercial initiatives. I will mention only two of the most significant attempts.

The first such major attempt was to form Telenet, a separate corporation with the goal of offering commercial network services to the nation. In our local version of the military/industrial complex revolving door, BBN hired Larry Roberts to serve as the first president of Telenet; Larry was the official at ARPA who had initiated and managed the ARPANET contract for the government. The decision to form Telenet was difficult, and a few people at BBN who were involved in network activities were so sure it was a good idea, and so annoyed with BBN's delayed decision, that they left to form a start-up company to provide such network services commercially. This staff departure probably played a minor role in encouraging BBN to get moving with the formation of Telenet. The establishment, operation, and growth of Telenet became a considerable drain on BBN's management attention, but this commercial initiative seemed promising for a number of years. Eventually, the capital requirements and other complexities associated with running a common carrier became difficult for BBN, and Telenet was sold to GTE.

The second major attempt at developing a commercial networking initiative occurred many years later. BBN decided to attempt acquisition and commercial operation of several of the key NSF regional networks, with the hope, once again, of providing a nationwide network service on a commercial basis. The university consortiums had begun to realize the long-term difficulty of network operation and were willing to consider such commercialization but, in some cases, under complex conditions and at high prices. BBN did acquire several such regional nets, and consolidated them into an entity that came to be known as BBN Planet. For a time, the plan looked promising, but it was an aggressive plan, requiring significant capital and other resources, and BBN was finally unable to proceed within the available BBN resources. This led to the eventual sale and breakup of BBN and harm to many people, including loss of jobs and dislocating transfers within BBN and to other companies.

7.5 Commercial imperative

Although the network-related commercial activities were large and important, they were by no means the only such commercial ventures. From the day I joined BBN until the day I left, BBN was immersed in a constant tension between operating a contract-based research and development organization and attempting to commercialize some of the developments of the R&D activities.

This tension between BBN's R&D and commercial thrusts had a number of sources:

- Individual scientists and mid-level managers were justifiably proud of some of the research results, believed that the world at large could use such results, and felt that both the corporation and they, as individuals, could benefit financially from commercializing those results.
- The R&D business, especially for the government, had a low profit margin, with pre-tax profits varying from 0 to 7 or 8 percent. Therefore, the company's top management was always interested in ways to take advantage of the research results in a manner that would increase profit margins. Further, the company's top management was adept at finding outside funding and partners for possible product activities, and perhaps enjoyed exercising those particular muscles.
- This was particularly true in the case of network activities and in the case of various software products where, for example, the ARPANET's expansion led to a demand for additional equipment at new locations. It was also true in the case of various software products that arose in the course of government-supported research and development.
- Although the R&D business had a number of attractive features, it also had some negatives: the constant need to seek out, market, and negotiate new contracts, the various government rules and regulations surrounding such research, occasional issues of classified work, and reporting requirements, for example. The lure of a product business with an income stream based on repeat sales and without the negative features of the government R&D business was very tempting to some staff and some managers.
- As a public company, the top management, employee stock-option holders, shareholders in general, and the financial community were concerned with the stock price, and thus interested in the possible upside growth of the stock price based on commercial initiatives.

These tensions led to a wide variety of commercial initiatives, some small ones internally funded and some large enterprises with outside funding or outside partners. (For a thorough list of BBN's commercial activities, see "The History of Technology Transfer at BBN" by Stephen Levy in this issue.) In my early days at BBN, these activities included commercial fuel cells; marketing of accelerometers and other measurement instruments; a West Coast division marketing various computer I/O devices; and a commercial time-sharing service (called Telcomp in the United States and Time Sharing Ltd. in the United Kingdom.

I particularly remember BBN's later commercial activities where technology and people left my R&D division; some examples are the Telenet and BBN Planet already mentioned, an effort to sell a multiplatform email system, a large software products activity, a network product and systems business, an attempt to exploit speech-understanding technology, and a program to exploit multiprocessor computer technology. About the time I was leaving BBN in 1994, the remainder of these commercial activities were being sold or folded back into BBN's R&D activities, with the exception of BBN Planet, which BBN was trying to expand dramatically.

As a high-level middle manager at the company, I was certainly complicit in the pursuit of some of the commercial initiatives, although I was seldom in full control of any of them. Thus, to the extent that most of the commercial initiatives, in my view, were insufficiently successful, I must share some degree of responsibility. "Insufficiently successful" is a strong term and requires some explanation. Many of the commercial activities hung on for a long time, simply did not make enough money, and were eventually closed or merged into another company unit. Some commercial activities provided a good return to BBN and/or outside investors and partners, but didn't seem promising for the long term. Some of the larger commercial activities at some point needed more capital than BBN was prepared to make available and were sold for a reasonable price. Some of the commercial activities did make money for BBN for a long time and then ran into trouble. But, in sum, none of the commercial activities ever grew into a long-term stable source of major profit for BBN, and certainly none of the commercial activities ever was a source of significant funding back into BBN's R&D groups.

The company's top management felt that, to the extent feasible, the commercial initiatives should be separated from the mainstream contract R&D business, and this was the course followed with all major commercial initiatives. This approach had a few hazards: The R&D groups, or their managers, didn't always want the commercial initiative torn away from them and managed separately; the transfer of people from the R&D groups to staff the new commercial initiative was often painful, either to the people moving or to the research program and morale of the group left behind, or both. The accounting, contract legalities, and reward structure issues associated with such splitting off of commercial activities were often difficult and time consuming.

There was another difficulty associated with this constant mix of contract research and development with commercial activities: The BBN management information system (MIS) for accounting was sorely strained by the company's overall complexity, and this severely drained corporate resources, both financial and personnel, over time. The responsibility for the MIS group actually moved back and forth between BBN's top financial management (who were responsible for the numbers and thought they knew what they wanted) and BBN's technical management (who thought that they understood the computer issues needed to produce the right numbers). Also, people with an MIS background tended to think "big machines" and "IBM" while the cost/performance of other smaller-machine approaches was more appealing to the technical groups who worked on the problems.

7.6 Diversions

In any institution, and certainly in any public corporation, a variety of events divert attention from the normal stream of R&D activity. Reorganizations, layoffs, or significant changes in procedures all cause some morale shifts. BBN was no exception to these hazards. However, several unusual diversions are noteworthy.

The first I will label the "guilty until proven innocent" diversion. Around 1978, BBN managed to infuriate a government auditor, and the U.S. Justice Department initiated a criminal investigation of certain BBN accounting practices. During this investigation, the government alleged that BBN as a corporation, and two BBN officials individually, were guilty of various accounting rule violations. The government issued a subpoena requiring the assembly and delivery of significant amounts of paperwork; the company was required to obtain significant legal assistance; and a number of BBN staff, although not targets of the investigation, were interviewed, some of whom needed to obtain legal help and were faced with testifying to a grand jury. The case was resolved by BBN and the two individuals, pleading guilty to some portion of the allegations. Consequently, a negotiated settlement was reached involving significant financial considerations, changes in the jobs of the two targeted individuals, changes in company practices, and promises regarding future compliance, among other things.

The whole affair, however, had a fascinating twist. In my view, BBN was not very guilty of much that mattered; neither the individuals nor the company stole any money, and the government at all times received more than fair value for the costs it incurred. Thus, in the "normal" judicial system that we think we have in the United States, where one is presumed innocent until proven guilty, BBN (in my view) should have, and would have, contested the charges with vigor and (in my view) might well have prevailed in court.

This stance was simply not possible — BBN was forced to plead guilty, or the firm would have gone bankrupt and closed — because, at the same time that the Justice Department was pursuing the case against BBN, the Defense Department (BBN's crucial client) had a rule that, in effect, said, "Well, we don't know whether you are guilty or innocent, but however long it may take the courts to find that out, the Defense Department cannot give you any new contracts or contract renewals." Further, similar rules required that other government agencies would have had to follow the Defense Department lead. This result would have put BBN out of business. So, it was crucial to settle the matter quickly, and this led to the guilty pleas. BBN's management deserved considerable credit for arranging to settle the case in a manner that avoided anyone's going to jail or a disastrous cutoff of government contracts. It was quite a diversion while it lasted, and it made me realize that "innocent until proven guilty" is only a catchy phrase and not necessarily how the judicial system actually works. After the settlement, BBN spent considerable time and money to ensure that government accounting practices followed the letter of the law.

The second unusual diversion, which I'll label the "quality diversion," was a bit more positive, but still caused considerable turmoil and expense. In the late 1980s, the common wisdom in the United States was that the Japanese were about to eat our lunch and that we had better learn what we could about how they were doing it. One approach that had been exploited in Japan was Total Quality Management (TQM). Basically, it is an attempt to carefully analyze how various activities are accomplished, to set goals, to involve all participants in the activity's analysis, and it is hoped, to greatly increase the quality of the result. Both in Japan and in the United States, the approach is most easily understood in a manufacturing or service activity, although proponents would claim that essentially any activity could be improved by using the approach. BBN's top management became very interested in TQM, and decided to try applying the approach to the entire company — to, in fact, attempt to make BBN a model of success in using this approach to improve performance.

TQM required widespread training sessions, in which we learned a variety of detailed methods for analyzing activities and detailed methods for considering changes in the activities, then noting the results. I was always dubious about TQM's applicability to BBN's R&D groups and viewed TQM-related activities as an expensive diversion. Although TQM activities were always charged to appropriate jobs or accounts, for me, the most telling symbol of an effort I viewed as misguided came when we were specifically instructed to avoid any attempt to segregate and thus track the costs of implementing TQM — this, despite the fact that most other identifiable activities were carefully monitored. Well, TQM is still alive and well in the United States, and it is no doubt helpful in many contexts, but luckily (from my viewpoint) after a year or two, the TQM effort at BBN was phased out, and most groups could get back to actual work. Also, after a few years, people no longer expected Japan to eat all of our lunch.

A third group of diversions was certainly intended to have positive results; it was an attempt to expand overseas. The most enjoyable example for me personally was in connection with "Silicon Glen" in Scotland. The Scottish Development Authority presented a convincing case that a company like BBN should open an R&D office in Scotland, in an industrial zone near Edinburgh where other U.S. companies had located Scottish branches. We became convinced that it made sense, and, after some effort, located a European manager for the office, leased space, and sent one of our young middle managers (who knew a great deal about the BBN culture) over to Scotland to serve as assistant manager of our new office. This office provided the benefit that a number of pleasant trips became possible to visit the new office, and we were all quite enthusiastic about our European experiment. Unfortunately, while the office did secure some work in Europe, it mostly was forced to rely on tasks subcontracted from our home office, and it was eventually closed. It had turned out that, while BBN was well respected, most potential European sources of contract work were not so interested in sending jobs to a rather expensive American firm, even an American firm with a capable and pleasant European office manager.

One interesting overseas venture was forced upon the company. We were obtaining SUE minicomputers from Lockheed Electronics in connection with a multiprocessor project called the Pluribus. The particular components we needed were manufactured by Lockheed in a small Hong Kong factory and, as I remember, were the only remaining items being manufactured at that Lockheed location. Lockheed Electronics fell upon hard times and, among other changes, decided to close the Hong Kong factory — with the potential result that BBN could no longer obtain the minicomputers we needed. So, despite the diversion of dealing with a small faraway activity, BBN bought the little Hong Kong factory. The factory was managed by an expatriate American, with at least modest competence in Chinese, who lived the life of the long-lost British aristocracy — when not at his house overlooking Resolute Bay, (think *Love Is a Many-Splendored Thing*), he might be at his club, and he did visit the factory to talk to his Chinese assistant manager. He was paid extraordinarily well by Lockheed for this lifestyle, and BBN had no choice but to continue his compensation. With him, the factory would run, maintain its lease, keep its employees, and so on; without him, we would not have any minicomputers. He was a very nice guy and, in addition to running the factory efficiently, he provided good tour services for visiting BBN management. BBN eventually closed the factory when there was no longer a need for the SUE minicomputers.

7.7 Closing comments

First and foremost, BBN was a great place for a technical person to work, and most people really liked working there. It was a middle ground between academia and the commercial world, with the meritocracy and individual freedom of the academic world, along with the potential reward structure and potential impact on the world of commercial ventures. In the case of the ARPANET and the subsequent explosion of network activity, it was an extremely unusual opportunity for a technical person to "ride a rocket" of change in the world.

Second, because the ARPANET project was so successful, it is worth a few words to consider why:

- At BBN, the project was operated with a small group of very talented people. All the hardware people could program, and all the software people understood a good deal about the hardware details.
- In the government, the people managing the project were as smart as or smarter than the people at BBN. Larry Roberts in particular was very bright and had the right mix of hands-off control with close attention to detail.
- The fledgling network needed users, and the various early sites at universities were not necessarily eager to connect local computers to a network and allow users from other places to use local machines. It was helpful that ARPA, through Roberts, was also funding these user sites and could exert early pressure for cooperation. Later, when the network had proved its utility, such concerns were less prevalent.
- Despite being a government and a Department of Defense project, ARPANET was entirely unclassified. Further, there was no restrictive access control or usage accounting, and the network usage was provided as a "free good," avoiding the necessity for people to make difficult cost/benefit decisions about trying the network.
- The network could adopt the transmission protocols that seemed to make sense, without our worrying about backward compatibility with the rest of the communications world.
- The contractual relation, on a cost-plus-fee basis, was amazingly free of the normal bureaucratic nonsense that often afflicts government contracting.

Third, it was discouraging that so many of the commercial ventures did not live up to expectations, and it probably indicates the difficulty of mixing research and development with commercial activities. Although BBN tried to separate the two kinds of activity, there was constant tension and interaction. Unfortunately, BBN's commercial activities were not blessed with the same luck as was historically found by the R&D groups.

Finally, it was quite amazing that an R&D group, without much corporate support from product activities, could flourish over many decades, serving the best interests of the goernment sponsors, the staff, and in general the common good. Even today, portions of the early BBN survive as a research institution, still serving such interests. Very recently (2004), a sizable research component of BBN became a venture-capital-funded separate corporation, and I wish it good luck in the future.

References

1. H. Goldstine, review of "Whirlwind: The History of a Computer Pioneer" by K. C. Redmond and T. M. Smith, *Annals of the History of Computing*, vol. 3, no. 1, 1981, pp. 74–77.

2. B. Randall, *"The Colossus," A History of Computing In The Twentieth Century*, N. Metropolis, J. Howlett, and G. C. Rota, eds., Academic Press, 1980, pp. 47–92.

3. *IEEE Annals of the History of Computing*, special issue on the SAGE system, vol. 5, no. 4, 1983, pp. 319–403.

4. *IEEE Annals of the History of Computing*, special issue on automating libraries, vol. 24, no. 3, 2002, pp. 59 and following.

5. BBN Annual Report 1967, BBN, Cambridge, Mass., 1967, p. 9.

6. K. Hafner and M. Lyon, *Where Wizards Stay Up Late*, Simon and Schuster, 1996.

Part III
Applying Computer Technology

This part of this volume contains a series of papers on more or less specific areas of application of computer technology:

- psychology
- control systems
- acoustic signal processing for detection
- DataProbe and ADCAP
- medical applications
- speech processing
- natural langauge understanding

Chapter 8

Psychology at BBN from the mid 1960s

Raymond Nickerson and Sanford Fidell

Chapter 3 by John Swets covers the beginning of psychological research at BBN. This chapter covers the period since then into the 1990s.

8.1 Introduction

Psychology at BBN goes back to the relatively early days of the company's history. J. C. R. Licklider, a Harvard and MIT psychologist already well-known for his work on hearing and psychoacoustics, was brought to BBN by then president and CEO Leo Beranek in 1957. Other chapters in this volume, especially those of Beranek and John Swets, note the enormous impact that Licklider had on BBN in several ways, not the least of which was through the several people that he, directly or indirectly, brought there. Shortly after arriving, he recruited Karl Kryter, W. Dewey Neff, and Vincent Sharkey; over the next few years, and largely due to Licklider's influence, this group was joined by Thomas Marill, Jerome Elkind, Swets, David Green, Richard Pew and John Senders.[a]

Psychology was well established at BBN by the mid 1960s. An account of the role that psychology played at BBN before that time, and how it relates to what was then going on in psychology more generally, is given by Swets in Chapter 3 in this volume. In this chapter we pick up the account of psychological research at BBN where Swets's leaves off, and cover the period from the mid 1960s into the 1990s. We describe how psychology fit within the organizational structure and its connection to acoustics and computer science and technology. We identify major contributors to the work during the period covered. We describe two computer-based laboratories that were used to conduct many of the psychological experiments done at BBN during that time. We attempt an extensive, though not exhaustive, account of the many projects that were undertaken. Much of the work is documented in journal articles, book chapters and books — as well as in BBN technical reports — and pointers are provided to many of these sources. We realize that the descriptions of projects may be of greater interest to many readers than the information about organization and operations, but the latter is important to the story of psychology at BBN and its relationship to computer technology, and it seemed natural to us to cover it first. Some readers may wish to skim

[a]The first author's attraction to BBN was largely due to encounters with Licklider while visiting there for a few weeks in the early 1960s in order to learn how to program a PDP-1 computer that had been acquired by a psychological research lab at Hanscom Air Force Base, Bedford, MA, where he was working at the time. An invitation to explore the possibility to join BBN later came from then Principal Scientist Senders, who had himself joined at the invitation of Licklider. Senders came to know Licklider as a student in a mathematical statistics course the latter taught at Harvard University in the 1940s; Licklider later became his honors thesis advisor. Senders was recruited by Licklider to join BBN, which he did in 1963 (shortly after Licklider had left), after working with the Air Force's Aeromedical Laboratory and setting up a human factors group at Minneapolis-Honeywell. Nickerson was hired at BBN in 1966 into a division headed jointly by Elkind and Swets, both of whom had been brought there by Licklider. By this time Licklider had left BBN for a stint at ARPA, but his influence was apparent everywhere.

the first few pages or go directly to the project descriptions, which are in the section with that heading.

In his chapter, Swets traces the path that BBN took from acoustics to behavioral science to computers. Happily for the acousticians and psychologists, the introduction of each new interest did not result in abandoning the interests that already existed. Quite the contrary, acoustics remained a focus of research and development at BBN long after behavioral sciences came along; similarly, acoustics and behavioral sciences remained major areas of activity after computers made their appearance on the scene. Each introduction of a new interest opened new opportunities for work in those that already existed. Many projects drew significantly from all three areas; it was BBN's mix of capabilities in the three domains that made the company unusually well qualified to undertake certain types of projects.

The parallel activities in the three major areas provided opportunities for cross fertilization and sprouted many interdisciplinary projects. Acoustics and psychology were combined early in the company's history, as Swets recounts. The interaction between psychology and computer science was evident in projects in artificial intelligence, computer interface design, educational technology, and user-oriented system design and evaluation. Computer-based speech processing and generation, and the measurement and prediction of the audibility and annoyance of transportation noise, are illustrative of work that required expertise in all three areas.

BBNers also did a considerable amount of psychological work that did not relate to computers very directly. The work that related most directly to computer technology did so in either of two ways. Some of it was aimed at influencing the design and development of computers or computer-based systems, but the larger portion applied computer technology to other ends. This distinction is not a sharp one; many projects included work of both kinds. Much of the psychological work at BBN was affected by computer technology even when it was not aimed at influencing the future development of that technology.

Many of the projects on which BBN psychologists worked are described in other chapters in this volume. These include especially projects in educational technology, speech and language, and artificial intelligence. Projects that are discussed in other chapters are not described in any detail here, although some are mentioned and the other chapters are cross referenced as appropriate.

8.2 Psychology in context at BBN

Organizational context

Most BBN psychologists were members of either the Experimental Psychology department or the Psychoacoustics Department, two of several departments within what was known at the beginning of the period of interest as the Behavioral Sciences Division and, as of 1975, the Information Sciences Division — one of the company's three major divisions at the time.[b] The Experimental Psychology Department was located in Cambridge, MA. The Psychoacoustics Department was located first in Van Nuys, CA and

[b]Other departments in the division and their managers as of the mid 1970s were Artificial Intelligence (William Woods), Control Systems (Sheldon Baron), Distributed Information Systems (Robert Thomas), Educational Technology (Wallace Feurzeig), Interactive Systems (Jerry Burchfiel), Sensor Signal Processing (Richard Estrada), Speech Signal Processing (John Makhoul), and the Research Computer Center (Theodore Baker). Nickerson was the director of the division from 1969 to 1984; Baron and Burchfiel were associate division directors beginning in 1975. In 1984, the Computer Science Division and Information Sciences Division were merged as the Computer and Information Sciences Division with Frank Heart and Nickerson as the director and deputy director respectively.

later in Canoga Park, CA, both of which are within the greater Los Angeles area. For convenience, we sometimes refer to its location as Los Angeles, except when the more specific designation is germane in the context.

The experimental psychology and psychoacoustics departments

In this chapter, footnotes are indicated by superscript letters; superscript numbers identify references located at the end of the chapter.

At the beginning of this history, the Experimental Psychology Department had five senior members (Allan Collins, Glenn Jones, Joseph Markowitz, Nickerson and Swets) and was managed by Markowitz, who was hired by Swets to help staff a NASA/Ames project (about which more later), shortly after getting his Ph. D. from the University of Pennsylvania. Markowitz worked on problems of signal detection,[1] reaction time[2] and vibrotactile perception[3]; he left BBN a few years after this story begins.

Thomas Triggs, who joined BBN in the late 1960s, managed the Experimental Psychology Department from 1969 to 1973. He was very active in the Human Factors Society (now the Human Factors and Ergonomic Society) at both local and national levels. His research while at BBN included work on design criteria for visual displays for aircraft and space vehicles[4] and on projects involving the use of computer-based displays in army tactical intelligence operations (see below). With Ronald Pickett, another psychologist who joined BBN in the early 1970s, he organized an international conference, held in Lisbon in 1974 (after Triggs had left BBN to return to his native Australia) under the sponsorship of the Science Committee of NATO on the topic of Human Factors in Health Care. The proceedings were published as a book with the same title the following year.[5]

Pew[c] became the manager in 1975 and remained in this capacity over the most of the remainder of the period covered in this chapter. He led numerous projects as a BBNer and contributed to many more, several of which are mentioned below. While at BBN Pew served as president of the Human Factors Society and as the first chairman of the National Research Council's Committee on Human Factors.[d] His service to his profession has been recognized with many honors, among the more recent of which was the naming of the Richard W. Pew chair in Human-Computer Interaction at the University of Michigan. He was appointed a BBN Principal Scientist in 1976. By the early-to-mid 1980s the Experimental Psychology department had grown to 12 to 15 doctoral level psychologists and a comparable number of support staff. Long-term members of the Experimental Psychology Department — BBNers for 10 or more years — included Marilyn Adams, Allan Collins, Carl Feehrer, John Frederiksen, Barbara (Noel) Freeman, David Getty, A. W. F. (Bill) Huggins, Glenn Jones, Daniel Kalikow, Nickerson, Pew, Pickett, Anne Rollins, Ann Rosebery, William Salter, Albert Stevens, Swets, Yvette Tenney and Beth Warren. Some of these people transferred into a different department when the predominant focus of their work made that appropriate.

The Psychoacoustics Department, was considerably smaller than the Experimental Psychology Department. Its manager was Karl Pearsons from 1968 to1982 and Fidell thereafter to 2001. Pearsons joined BBN immediately after graduating from M.I.T in 1956 and remained with the company for 45 years, working on numerous noise mea-

[c]Pew did three different stints at BBN. He came first in 1958, at the invitation of Licklider, after separation from the U.S. Air Force. He left in 1960 to pursue a PhD at the University of Michigan, where he remained as a faculty member after receiving his degree. He came back to BBN to spend a sabbatical leave in 1970-71, returned to the university, was recruited back to BBN in 1974 and has remained there since. He maintained a tie with the University of Michigan, however, and has led a short course in Human Factors Engineering there every summer since 1965. This course has been attended by approximately 2500 people, most of whom are professionals working either in human factors engineering or some closely allied field.

[d]Most NRC standing committees and boards come and go; the Committee on Human Factors celebrated its 30th year of existence in 2010, at which time it became the NRC's Board on Human-Systems Integration.

surement and control projects often collaborating with physicists, acoustical engineers and psychologists alike. Fidell was hired by Swets into the Van Nuys office upon graduation in 1968 at the recommendation of Wilson P. ("Spike") Tanner, Jr., Swets's mentor at the University Michigan and the chair of Fidell's Ph. D. committee. Other long-term members of the Psychoacoustics department were Ricarda Bennett, Richard Horonjeff, Richard Howe, Laura Silvati, Matthew Sneddon and Suyeo Tomooka. The work of members of both of the Experimental Psychology and Psychoacoustic Departments is mentioned in what follows and in other chapters in this volume.

Departmental operation

In many respects, each department functioned like a research-oriented university department, without the responsibilities of teaching and academic committee work. Support for projects was obtained primarily from government agencies, corporations, trade associations, and private foundations, as a result of proposals written by senior members of the departments. Projects typically were managed by the people who were responsible for obtaining the support for them and staffed according to the particular projects' needs. It was commonplace for the members staffing a project to have indicated an interest in working on them as they were being conceived, and often to have contributed to the proposals. Departmental organization remained relatively unchanged over time; project organization and management, in contrast, changed regularly with the requirements of the projects as they came and went. Department boundaries were an organizational convenience, highly permeable and not barriers to project staffing; the members of any given department participated freely as needs and opportunities dictated with people all over the company.

Parallels between BBN departments and departments at research-oriented universities were many. Chapter 5 describes them in some detail. The Experimental Psychology and Psychoacoustics Departments illustrate this correspondence very well. Members regularly published in technical journals and gave papers at professional society meetings; served as officers in professional organizations, as chairs or members of National Research Council committees, as journal editors and members of journal editorial boards, as members of standards organizations such as ANSI and ISO, and on Ph. D. committees of various universities; in general, they engaged in the same activities as did their university colleagues. The major exception was that most did not teach, but some even did this on a part time basis, giving courses as a BBN employee at a local university or lecturing in BBN's Program of Advanced Studies. (See chapter by Levy.) For several years, the Experimental Psychology Department hosted a Memory and Cognition Seminar, which met monthly and drew participants from Harvard, MIT, Brandeis, Boston University and other Cambridge-area universities. From time to time the wider psychological community in the Boston-Cambridge area was invited to talks of interest to that community by visitors to BBN, including those sponsored by the Science Development Program's Guest Lecturer series (See BBN Culture in this volume.)

One way in which BBN departments differed from their counterparts in universities was with respect to availability of research grant support. BBN only rarely obtained grant funding. As a profit seeking organization, it functioned in the domain of contracts. Contracts differ from grants most notably in terms of expectations with respect to deliverables and schedules. As a general rule, contracts are awarded for the procurement of specified products — which, for a research organization, may be reports, computer programs, or other types of software as well as hardware devices or systems. Oversight and control are generally tighter and written reporting of progress on project objectives is typically required at regular intervals, sometimes as short as monthly. Perhaps the most negative aspect of this difference to researchers who liked doing basic as well as applied research was the difficulty this represented for obtaining funding to do the kind of basic research that is typically done in university laboratories. Some BBN psychologists managed to do some relatively basic research, but support for it was not easy to obtain; our costs tended to be high relative to those of universities and, as a profit-seeking company, we asked for a fee.

Over the years, many university faculty members worked at BBN either as part-time employees or as consultants. We mention here only a few of those who participated significantly on psychology projects. Long-term associates of BBN include Professor David Green (while at the University of California in San Diego, when working with the Los Angeles group, and while at MIT and Harvard, when working with the Cambridge staff), MIT Professors Kenneth Stevens and Denis Klatt, and Professor Barbara Tabachnick at California State University in Northridge. Green was instrumental in helping establish

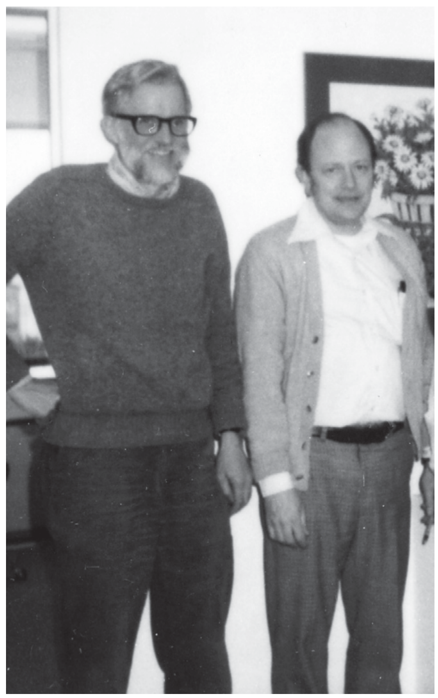


Figure 8.1 Ken Stevens and David Green.

the psychoacoustics laboratory in Van Nuys, CA, and a key participant in the early work in that facility. Stevens and Klatt participated in many projects involving speech in one or another way (See chapter on Speech Processing in this volume). Stevens was central to work on computer-based speech-training aids for deaf children. Other university faculty that played significant roles in psychology projects included Professors Richard Herrnstein and David Perkins from Harvard, who were major contributors to Project Intelligence and Douwe Yntema from MIT who participated in projects involving human-computer interaction. Tabachnick assisted with statistical modeling of sleep disturbance, annoyance, epidemiological, and property value data. Several of the projects just mentioned are described in more detail later.

Cultural context

In both the popular media and in the technical literature, one encounters distinctions between science and engineering, between research and development, between basic and applied research, and so on. The Department of Defense has a system for categorizing research and development activities that recognizes a continuum going from basic (6.0) research to system development and deployment (6.4). Individual BBNers worked at all points of this continuum, sometimes serially and sometimes simultaneously.

It is fair to say, however, that there was a tension — a competition of sorts — between people or groups who identified more with the research end of the spectrum and those who were more focused on engineering and system development. We do not mean to overstate this tension, especially in view of the fact that many BBNers had, by preference, a foot in both worlds, but to fail to acknowledge its existence would be to omit an important aspect of the culture.

One way in which the tension found expression was in preferences for the kinds of activities the company would support with limited discretionary funds. Those on the research end of the spectrum tended to want support for the writing of papers or books, the convening of conferences or seminars, the provision of sabbatical leaves for Principal Scientists, and the like, whereas those on the system development end tended to prefer the company to support the designing of a device that might lead to a patent, the development of a business plan for an entrepreneurial venture, or the construction of a prototype system that could demonstrate an engineering concept.

Most BBN psychologists tended to identify more with the research end of the spectrum. This is not to suggest that they lacked interest in seeing the results of their research applied to practical ends — to the contrary, most were very keen to have them used to good effect — but they attached considerable importance to publishing, conferencing and other activities that people who identify more with engineering and system development might tend to view as predominantly academic.

We think this tension, which, at the risk of gross oversimplification, we might refer to as a tension between research and development, was beneficial to the company, because it assured a balance that made BBN the stimulating and productive work environment that it was. If either the research or the development faction had dominated the other, BBN would have been a very different, and in our view, much less interesting, place.

8.3 Overview of psychological studies at BBN

The work in psychology at BBN covered a broad range of problem areas. At any given time there were about a dozen projects ongoing that were primarily psychological in nature. These projects varied in size from one- and two-person efforts to those requiring teams of a dozen or more. Sponsors of the work included the Department of Defense (Army Research Institute, Army Tank Command, Air Force Office of Scientific Research, Aerospace Medical Research Laboratory at Wright-Patterson Air Force Base, Naval Training Equipment Center, Office of Naval Research, Advanced Research Projects Agency), the Federal Aviation Administration, NASA (Langley and Ames), the Social Security Administration, the U.S. Department of Education (National Institute of Education, Bureau of Education of the Handicapped), the National Institutes of Health (National Cancer Institute, National Institute of Child Health and Human Development, National Institute of Neurological Diseases and Stroke), the Departments of Agriculture (U.S. Forest Service) and Interior (National Park Service), and the Internal Revenue Service.

Most of the projects on which psychologists and human factors engineers worked at BBN can be partitioned roughly into four types:

- Small to medium-sized projects under contracts obtained by submission of unsolicited proposals to government agencies or foundations for field-initiated research.
- Projects supported by contracts obtained by competitively bidding in response to Requests for Proposal issued by government agencies.
- Interdisciplinary projects involving teams composed of specialists in a variety of areas, psychology or human factors engineering among them.
- Collaborative projects in which BBN served as a subcontractor to a university or other research organization.

Although the psychological work was diversified and the mix of projects varied over time, it is possible to identify some themes on which work continued for relatively long periods. These include highway safety and driver performance, human-computer interaction and system design, teaching and learning, the role of computers in education, utilization of medical imaging, assessing community response to environmental noise exposure, and predicting the audibility of sounds propagating long distances outdoors.

Funding was, of course, a constant concern. We lived on what universities refer to as "soft money." As already noted, almost all of the psychological projects undertaken at BBN were done under contract with a government, non-profit, or commercial agency or organization. Unlike at many industrial laboratories, there were relatively few company-sponsored projects. Although many of our projects were "follow-ons," which is to say they were for agencies with whom we had worked before, relatively few of the contracts were for more than one year.

Diversification was also a constant goal; we were always on the look-out for opportunities to extend our project mix and broaden our support base. Indicative of this interest was our hiring of an individual — Paul Horwitz, who had a PhD in nuclear physics, and had been a congressional fellow — explicitly to take the lead in helping find and pursue new funding opportunities, especially involving the human factors of nuclear power plant control room design and operation. Horwitz worked very hard at this and we in fact did get a project with the Electric Power Research Institute (about which more below), but he soon became a major contributor to projects on educational technology, for which he also had a passion and a rash of creative ideas. Some of his work in this area is described in the chapter in this volume on that subject.

Essentially all projects produced one or more technical reports for the sponsor. In addition, however, many, if not most, projects yielded publications in the open archival literature: articles in professional journals, chapters in edited books and handbooks, proceedings of professional conferences, and authored books. Interest in publishing in the open literature is one of the traits that BBN psychologists shared with their academic colleagues. Many BBNers were highly self-motivated to publish, and the BBN management encouraged this interest. In 1989, the Science Development Program (See BBN Culture) began recognizing outstanding publications explicitly by establishing annual $2000 awards for each of the best publications — as judged by a committee of Principal Scientists — in each of three areas, physical sciences, life and social sciences, and computer and communication sciences. At the same time it established an annual $2000 "young author's award" to go to the best paper, irrespective of category, published by an author under 35 years of age. Eventually the program was revised so as to provide cash bonuses for all articles appearing in refereed journals.

As another means of encouraging interest in publishing, a "BBN Authors' Bookshelf" was established in the library (see Figure 8.2) where books that BBNers had authored, edited or contributed chapters to were prominently displayed. (According to a memo,

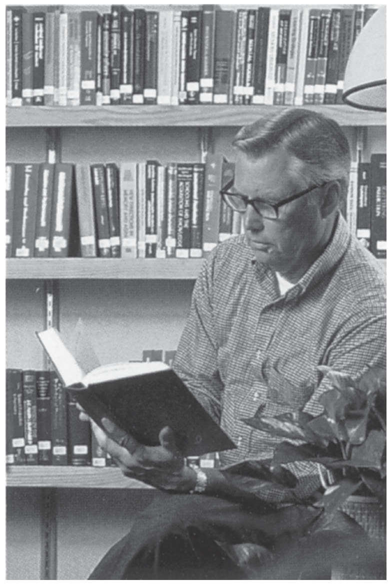


Figure 8.2. The BBN library's BBN authors' bookshelf featuring books written, edited, or containing chapters by BBN personnel.

dated 1 March, 1988, from Nickerson to BBN Laboratories Staff, there were, at that time, 160 books in this collection, but the memo was a request for information regarding books that should be added to the collection, which we knew to be incomplete.)[e]

Small to medium sized field-initiated projects

This research tended to be very similar to that typically done by researchers or research teams in university psychology departments. The idea for the research project almost invariably came from the Principal Investigator, who wrote the proposal for the project and did the work, perhaps with one or a few assistants.

Several of the projects described in this volume are in this category. Examples include the work on signal detection done with support from NASA Ames, work on speech training aids for deaf children done for the Bureau of Education of the Handicapped, and studies of teaching and learning done with support from the Office of Naval Research.

[e] We venture the guess that the output of BBN psychologists would compare favorably with that of any first-rate university psychology department of comparable size in the country. By way of giving this observation some credence, we note that Swets authored, or co-authored, in addition to the classic 1966 book on signal detection theory and psychophysics (with Green), several other books, six articles in *Science* between 1961 and 1988, the inaugural article in *Psychological Science in the Public Interest* a version of which was published also in *Scientific American* in 2000, numerous articles in major psychological and medical journals, and book chapters, many of which have been republished in anthologies. We do not mean to suggest that this level of productivity is representative of that of BBN psychologists generally, but it provided a standard to which others could aspire, and several did publish widely. Prominent among the journals in which BBNers published were: *Cognitive Psychology*, *Cognitive Science*, *Computerized Medical Imaging and Graphics*, *Human Factors*, *IEEE Transactions, International Journal of Aviation Psychology*, *Investigative Radiology*, *Journal of the Acoustical Society of America*, *Journal of Experimental Psychology*, *Journal of Sound and Vibration*, *Medical Decision Making*, *Noise Control Engineering Journal*, *Psychological Bulletin*, *Psychological Review*, *Psychological Science*, *Radiology*, and *Science*.

Competitively bid projects

Many of the projects undertaken by BBN psychologists were funded as the result of winning a competitive bid. In such cases a request for specific work came from the agency that wanted the work to be done, typically as an announcement in the Commerce Business Daily. Often the requested work was described in terms of objectives, and bidders were free to fashion the methodology they believed would best accomplish those objectives. And, of course, cost was always a major criterion against which bids were evaluated. We always felt that this put us at a bit of a disadvantage when competing with universities for projects — which we frequently did — because our cost structure did not compare favorably, for bidding purposes, with theirs; but we did bid nonetheless, and frequently won.

Among the competitively won projects that we had were one with the Army's Behavioral Sciences Research Laboratory (BESRL) on army tactical intelligence, one with the Naval Training Devices Center on decision making training, one with the National Institute of Education on the teaching of higher-order thinking skills, one with the Social Security Administration to develop a laboratory to facilitate the introduction of computers into its field operations, one with NASA on aviation safety, one with the Air Force to study the impact of sonic booms, and one with the National Park and Forest Services to study the effects of noise on outdoor recreation. There were others.

Multidisciplinary projects

Many of the projects on which psychologists worked at BBN drew on the expertise of people from a variety of disciplines, including physical acoustics, signal processing, and statistical analysis. In some cases, the psychologists served as consultants or resource people; in others in which the bulk of the work was psychological in nature, people from other disciplines played consultant and resource roles. Often, however, people representing a variety of disciplines all played major roles in what constituted truly multi-disciplinary work.

Collaborations

For several years (approximately 1977 to 1990) BBNers collaborated with the University of Illinois on the establishment and maintenance of a Center for the Study of Reading, which was sponsored by the National Institute of Education. The center was initially funded for five years on the basis of a bidding competition, which the Illinois-BBN team won. The same team twice won the recompetition five and ten years following the initial establishment of the center. The collaboration worked very smoothly over the course of the first two five-year cycles, but difficulties developed during the third that strained the University of Illinois-BBN relationship. Nevertheless, the long collaboration was highly productive and yielded an impressive stream of publications — Center reports, journal articles, book chapters and books — addressed to the question of how to improve the teaching of reading and the learning of same. More will be said about the center when individual projects are described.

Another major collaboration that also lasted for several years (approximately from 1979 to 1983) was with Harvard University on Project Intelligence, a project undertaken in Venezuela at the request of the Venezuelan government. The objective of this project was to develop a course, to be used at the seventh-grade level, to help students improve their thinking skills. This project, which was unusual in several respects, is also described in more detail subsequently.

BBN collaborated with Bank Street College of Education, Harvard University, and

Brown University on the Center for Technology in Education from about 1989 to 1994. This center, like the Center for the Study of Reading, was funded by the Office of Educational Research and Development and was one of about a dozen national centers focused on one or another aspect of education (reading, writing, teacher education) being funded at the time. The charter for the Center for Technology in Education was to explore ways in which technology could be used to improve educational practice. Some of the work done in this collaboration is described in the chapter on Educational Technology.

There were several instances of collaborative projects between BBNers and universities or hospitals — the University of Chicago, the University of Cincinnati, the Harvard Medical Schools, the University of Massachusetts Medical Center and the Brigham and Women's Hospital — in the area of medical imaging. Funded primarily by the National Cancer Institute, these collaborations varied in size and duration — those with radiologists at the University of Massachusetts Medical Center and Brigham and Women's lasted for over 25 years — and involved work on image-based diagnosis of cancer and the classification of cataracts.

Other projects involving collaboration between BBN psychologists and universities included one with the University of Illinois to develop training strategies (funded by DARPA), one with Harvard University on microcomputers and literacy (NIE), one with Lesley College to develop material for pre-college math and science instruction (NSF), one with the University of Chicago, Brown University and the University of Michigan to do research on survey techniques (NSF), one with the University of Pittsburgh and the University of Massachusetts to develop an integrated system to assess and enhance basic job skills (U.S. Air Force) and one with the University of Alaska on the use of computers to teach language arts (local school districts and the university).

With the exception of Project Intelligence and the Center for Technology in Education, all the projects mentioned above were ongoing during the BBN fiscal years 1985 or 1986. An informal survey of BBN-university interactions was taken at that time for company purposes, and the results of that survey are the basis of the information provided here. A survey made at a different time would have yielded a different, but probably not greatly dissimilar, set of projects.

Consulting and extracurricular work

BBN psychologists sometimes consulted on an ad hoc basis. A case in point involved the accidental shooting down in 1988 of an Iranian passenger jet (A300 Airbus) by a U.S. naval vessel, the Vincennes, which was engaged at the time, along with another U.S. Navy ship, in a battle with Iranian gunboats. Naval personnel mistook the airliner for a military aircraft and fired to prevent an attack, killing all 290 people on board. Shortly after the incident there was a congressional hearing on issues of human decision making relating to it. The American Psychological Association asked four psychologists to prepare testimony for the hearing, one of whom was Pew. Pew prepared testimony regarding human factors matters that could have contributed to the tragedy. Before the hearing he was briefed by the navy on the details of the incident and on certain design flaws in the AEGIS equipment the navy personnel believed to have contributed to it. Although the navy had sponsored research on human decision making long before the Vincennes incident, shortly after the congressional hearings, it began a major new initiative in decision making that included both a training component and a component on system design.

Consulting work was likely to tap expertise in human factors engineering and human performance in person-machine systems. Charles Dietrich, Duncan Miller and

Pew consulted with the legal department of the Ford Motor Company on problems in which drivers alleged that their cars were "popping out of park" and rolling backwards after they exited the car. This relationship resulted in BBN conducting an extensive set of experiments on how drivers shift automatic transmission autos. Pew, David Getty and A. W. F. (Bill) Huggins consulted for General Motors Corporation on a suit brought by the Department of Transportation that alleged that vehicles based on the X-Car chassis had a rear-brake lock-up problem that was exacerbated by individual driver behavior.

BBN psychologists were called on to chair or serve on various government and National Research Council commissions and committees.[f] Some wrote, edited, or contributed to reports for NRC committees that figured prominently in federal transportation noise policy. Many chaired or served on panels, task forces, advisory groups, forums, or other entities convened by various government agencies or professional societies/associations to address specific problems. Generally this work was pro bono and was considered part of the appropriate overhead costs of a research and development corporation. Some of it was sponsored by BBN's Science Development Program (See BBN Culture).

8.4 Intra-BBN connections

Especially notable among the various areas in which BBN psychologists worked are three that are featured in other chapters in this volume: educational technology, artificial intelligence and speech technology. Details about this work will not be given in this chapter when they are provided elsewhere. It needs to be said, however, that the dividing lines between areas are sufficiently fuzzy that in some cases the decision to discuss a particular project in one chapter rather than another was made somewhat arbitrarily.

Psychology and educational technology

Perhaps the greatest involvement of psychologists in work not highlighted in this chapter was in the area of educational technology.[g] The bulk of this work is described in the chapter on educational technology, but much of it could just as well be described under the rubric of artificial intelligence, inasmuch as it drew on, and contributed to, that area of research to a considerable degree.

Psychology and artificial intelligence

There were many other interactions also between people in the psychology group and those working on artificial intelligence. The collaboration between Collins and Ross Quillian in which they tested empirically some of the implications of Quillian's[6] "teachable language comprehender" as a model of human semantic memory has been widely cited in the psychological literature and the stimulus for much subsequent research.

[f] Swets served as chair of the NRC's Commission on Behavioral and Social Sciences and Education and of its Committee on Techniques for the Enhancement of Human Performance; both Pew and Nickerson chaired the NRC's Committee on Human Factors. Pew was the founding chair of this committee.

[g] This includes work on computer-assisted instruction and intelligent tutoring sponsored by DARPA and ONR, led primarily by Collins, John Seely Brown or Jaime R. Carbonell with major contributions from Adams, Nelleke Aiello, Madeleine Bates, Geoffrey Brown, Jaime G. Carbonell, Frederiksen, Laura Gould, Mario Grignetti, Mark Miller, Joseph Passafiume, Eleanor Warnock and Barbara White. It includes also projects on second language learning, sponsored by DARPA and led by Swets, and on speech training aids for deaf children, sponsored by the Bureau of Education for the Handicapped and led by Nickerson.

The development by A. Stevens, Bruce Roberts and their colleagues of a simulation of a steam power plant — Steamer, described in the chapter on educational technology — is another instance of work that falls in both the educational technology and artificial intelligence domains. The simulation incorporates a knowledge base about the operation of a steam power plant, and its intended use was for instruction in plant operation. Simulation offers the possibility of training in situations that would be dangerous in a real plant.

Many other projects involved both educational technology and artificial intelligence. A collaboration between Collins and Jaime R. Carbonell, which produced a "mixed-initiative" Intelligent Computer-Aided Instruction (ICAI) system called Scholar, is a case in point.[7] Initially funded by the Office of Naval Research, the work continued for several years after Carbonell's untimely death in 1973 and was expanded to include additional research on plausible reasoning with support from both the Office of Naval Research and the Army Research Institute.

Psychology and speech technology

Psychology was also involved in speech technology work, and especially in projects in which application of the technology in some practical setting was a goal. Questions of speech quality — intelligibility and naturalness — are important determinants of the acceptability of vocoded, synthesized or digitized speech, and the design of techniques to make such evaluations is a psychological problem. Huggins was a major contributor to work in this area, bringing to bear multi-dimensional scaling techniques similar to those used in the radiology studies described elsewhere in this chapter.[8]

Projects involving the application, or studies of the feasibility of application, of speech technology in practical contexts included the use of speech for controlling automobile devices, such as the radio, windows, and air conditioner or to obtain personalized news, weather, sports and stock reports from the Internet, and an Interactive Speaker Identification System (ISIS) that would help an analyst identify who was speaking on a recording.[9]

Getty, Tenney and Freeman provided human factors support for several applications of speech recognition or speech understanding in a variety of phone-system contexts. For example, they helped evaluate the efficacy of existing Interactive Voice Technology systems for Verizon and other companies by analyzing end-to-end calls (starting with the automated answering, punching buttons, etc. and ending with a conversation with a real agent) to determine whether they had routed themselves to the correct agent and gotten there efficiently. (The process used in this work led to a patent.) The same group has worked on creating and improving call flows for various Verizon systems (business, consumer, wireless, online).

8.5 Facilities

In his chapter in this volume, Swets describes how some of the seminal work done by Licklider and his colleagues on human-computer interaction and on educational technology made use of the PDP-1 computer that was acquired by BBN in 1959. The same computer was used by Swets and his colleagues to run a set of experiments on learning to identify complex sounds; this was among the earliest published psychological experiments to be run by a computer — possibly it was the first. (See Chapter 3.) Today computer-controlled experimentation is the norm; in the early sixties it was just beginning to become a reality.

From those beginnings, computer technology became the primary instrument for conducting much of the psychological research done by BBNers. Two facilities deserve special mention, each a laboratory designed to be used for experimentation on perception (especially auditory) and cognition, one in Van Nuys, California and one in Cambridge, Massachusetts.

The Los Angeles laboratory

As already noted, the "Los Angeles" Laboratory was originally constructed in Van Nuys, CA, and later reconstructed in Canoga Park, CA. It contained an anechoic chamber and a PDP-8 computer, augmented with a digital tablet, a drum plotter, 12-bit D-A converters, a teletype for a keyboard, and paper-tape I/O operating at 10 characters per second. Fidell was largely responsible for configuring the laboratory and ensuring its operation. He, Richard Horonjeff and Allan Paul (with the assistance of much critical comment from Green) did much of the original software development for the real-time control and adaptive experimental design applications; hardware assistance was provided by Tomooka, Ronald Burns, Oran Zitella, and Peter Costello, among others. Over time, the PDP-8 was interfaced with numerous other devices, making it an increasingly versatile facility for experimentation. A custom designed interrupt register was added to manage the attention demands of these devices.[h]

This laboratory was used extensively to conduct studies on aural detectability and perceived noisiness of various types of sounds,[10] the effects of noise on speech perception,[11] the noticeability of signals of varying signal-to-noise ratio,[12] and annoyance of sound.[13] Adaptive testing procedures were developed that increased the efficiency of data collection,[14] ensuring, for example, that the intensity of signals in a signal detection experiment would vary around just detectable levels. In addition to controlling experimentation — scheduling and presenting stimuli and recording responses — the computer was used to analyze the data collected on the premises as well as data collected in the field (e.g., recordings of aircraft overflights, logs of well-drilling operations for BBN Geomarine, traffic counts from photographic frames).

Studies of sleep disturbance from noise were conducted with the use of telephone line interfaces between the computer and participants located in their homes. One-third octave band analyzers interfaced to the PDP-8 and the software for analyzing aircraft flyover noise data provided the basis for an aircraft certification business when Part 36 of the Federal Aviation Regulations was completed in 1969. Part 36 required complicated calculations of duration-adjusted and tone-corrected Perceived Noise Levels, a procedure developed by Kryter and his associates in the Cambridge office in the late 1950s in connection with pioneering consulting work for the Port of New York Authority that enabled the start of jet-powered civil aviation in the United States. (Not least among the many uses of the Van Nuys PDP-8 was its support of lunchtime Space War games.)

Of course, computer technology moved very rapidly during the years following the establishment of the PDP-8 laboratory, but this machine remained in use long after others of considerably greater power had become available. The investment in data

[h]Lacking any operating system, the computer was programmed in assembly language (and occasionally directly in machine code). Although programmers became proficient at putting the binary loader into memory by hand, home-brew logic was soon built to re-load the binary loader from a button push. This was regarded as a major productivity-enhancing device and an undeniable convenience feature. A locally-designed "halt on program counter address" capability was likewise considered a major advance in debugging technology. The only software development tools available initially were a DEC-supplied editor and a three-pass assembler. Feeding rolls of paper tape into the teletype reader for a major re-assembly of a large program would take the better part of a day. Debugging was accomplished manually, either by single-stepping the processor, or by replacing strategically-located NOP instructions with halts, and then examining the contents of the accumulator and other program registers and memory locations.

reduction and laboratory control logic and software, as well as the large stock of spare parts and a knowledgeable technical staff, kept the PDP-8 in use well into the 1980s, long after it was technologically obsolete. In time, however, much of the work migrated to successor VAX-family and eventually PC platforms.

The Cambridge laboratory

The earliest uses of a computer to conduct psychological experiments in Cambridge involved the PDP-1, but this machine had many uses and over time there developed a need for a computerized Human Performance Laboratory dedicated to psychological experimentation. The Cambridge PDP-8 based laboratory was established in 1966 to support work for NASA Ames on signal detection and its application to human performance being done by Swets and Green. This lab served as the center for psychological experimentation in Cambridge for several years, not only on signal detection, but in other problem areas as well. However, in the mid 1970s, funding was obtained from the National Cancer Institute to investigate the relative effectiveness of various medical imaging techniques. The computing needs for this work motivated replacement of the PDP 8 first with a PDP 11/34, and later with a PDP 11/70.

Primary responsibility for designing and operating the PDP 11/70 lab, which supported especially, but not only, experimentation involving the interpretation of medical images was Getty's. Almost all the programming for the Human Performance Lab over the years of its existence was done by Getty, Freeman and Huggins. The first project to use the PDP 11/70 facility involved evaluation of the first computed tomography (CT) imaging system developed by EMI Ltd. A CT display workstation that emulated the EMI had to be built in order to conduct reading sessions with radiologists.

Among the experimental topics addressed with the Human Performance Lab were signal detection in complex visual displays, pattern recognition and classification generally, advanced techniques for graphic display of multidimensional datasets, spatial information processing, timing of motor responses, human memory, and perceptual processing of true volumetric 3D displays (SpaceGraph TM). Users of the facility included most of the BBN psychologists who were doing experimental work at the time. BBN's prominence in pattern recognition work was reflected in a symposium on the topic that was held at BBN in 1978; the symposium was organized and chaired by Getty and a resulting book, *Auditory and Visual Pattern Recognition*, was edited by Getty and James Howard of George Washington University.[15] The work on the evaluation of medical imaging techniques is described in several journal articles and book chapters, and in a book by Swets and Pickett.[16]

In the mid 1980s the PDP-11/70 was replaced with a time-shared PDP MicroVax II. But as PCs and desk-top Macs became increasingly available and versatile, more and more of the work that once required a centralized facility was transferred to these desk-top machines. By the late 1980s essentially everyone at BBN who had a use for a PC or Mac (which was nearly everyone) had his or her own machine and a direct line to the company's time-shared computing center; the need for the MicroVax diminished to the point that it was retired around 1990. The Human Performance Lab continued to be used to conduct experiments, but with the now adequately powerful personal machines as the driving engines.

8.6 Project descriptions

Here we provide some details regarding several projects that, in the aggregate, illustrate the range of problem areas in which BBN psychologists worked. Where possible, we provide pointers to publications in which further details can be found. The reference list is not exhaustive — some projects produced many documents — but it is extensive and should suffice to give the reader who wants more information a good start in finding it.

The range of problems on which BBN psychologists worked and the mix of project types make classification difficult. Any of the various ways in which we have thought of organizing the descriptions of specific projects has some degree of arbitrariness about it. Strictly as a matter of convenience, we have selected a few generic topics under which most of the projects can be placed, without too much forced fitting. We begin with projects that might be considered to be closer to the pure science end of the science-engineering or basic-applied continuum and proceed to some that dealt explicitly with the development of one or another type of system.

Psychophysics and perception

Signal detection. Work on *signal detection theory* and its application to human performance was done for several years under contract with NASA Ames. The first notable publication to come from this work was the classic *Signal detection theory and psychophysics* by Green and Swets.[17] Numerous other publications appeared in subsequent years, including long after completion of the NASA contract.[18] In 1968, Swets organized a conference for NASA on Applications of Research on Human Decision Making; participants included Earl Alluisi, Richard Atkinson, Ted Birdsall, Ward Edwards, Jerome Elkind, Lloyd Jeffress, Alfred Kristofferson, John Senders, Richard Shiffrin, and Douwe Yntema. Many other signal detection studies were conducted in the Psychoacoustics Department as discussed below.

Reaction time and pattern matching. When Nickerson first came to BBN, he worked on several projects and taught a course at Tufts Unviersity (under a BBN contract) before he had any funded research projects of his own. Swets generously made room for him on the NASA project to do some experiments on human reaction time.[19] Later the same facility was used to do additional work on reaction time and time estimation for the Air Force Office of Scientific Research.[20] Most of the programming for these experiments was done by Freeman.

Another line of experimentation that made use of the Cambridge PDP-8 laboratory involved visual pattern matching, or short-term visual memory. This work, like that on reaction time, was relatively basic research, aimed at improving understanding of certain aspects of how visual information is stored and used over short periods of time. The experimentation that was done was greatly facilitated by using the computer to generate visual patterns with specific properties, as well as to collect and analyze data on participants' performance of various matching tasks.[21]

Psychoacoustics of sound and noise perception. The first computer-based psychoacoustic study conducted in the Van Nuys office was a 1968 NASA-sponsored study of the noisiness of impulsive signals.[22] The matter was of considerable practical interest at the time, since public annoyance by sonic booms was seen as a major impediment to the operation of an overland supersonic transport fleet. A family of transient signals (an ideal N-wave with nearly instantaneous rise and decay times, an N-wave with a slower rise time, a triangular waveform, a square wave, and a doublet) was created with varying durations, frequency content, repetition rates, and phase spectra. The annoyance of these signals was judged in a fully factorial, adaptive paired comparison design against the annoyance of a one-second long sample of an octave band of white noise.

Describing this study in some detail serves to illustrate the usefulness of the computer in conducting studies of this sort. Such paired comparison testing had been manually controlled in a cumbersome, labor-intensive process until minicomputers became available to automate the procedure. Typically, a reel-to-reel tape with a single, fixed sequence of test signal pairs was prepared in advance of testing. On each trial, an experimenter would start and stop the tape recorder to play a pair of sounds, and record the listener's stated preference for the first or second of each pair. Since it was difficult for an experimenter to reliably adjust step attenuators between trials, the sound levels at which pairs of signals were presented for judgment were determined in advance of the start of testing. The resolution of the method was fairly coarse, because the risk of an incorrect guess about the point of subjective equality of judgment of the fixed and comparison signals was great enough that 5 or 10 dB differences were needed between signal pairs. Furthermore, individual differences in annoyance judgments were great enough that the test tape had to span a large range of absolute levels of test and comparison signals. All of these constraints adversely affected the cost-effectiveness of the testing method.

In most BBN studies, test sounds were presented for judgments by individual listeners via loudspeaker under free field listening conditions in a large anechoic chamber. The test signals were generated in real time by playing waveforms stored in memory through one 12-bit digital-to-analog converter, while the computer used another D/A channel to vary signal presentation levels by means of a voltage controlled amplifier. Mario Grignetti generated the test signals on a mainframe computer in BBN's Cambridge office. A Fast Fourier transform (FFT) was performed to produce frequency-domain information. This information was converted from real and imaginary components to magnitude and phase, and a new, random phase angle was assigned to each point. The inverse FFT was then calculated to yield signal waveforms with different phase spectra but identical power spectra. In those days prior to wide area computer networks and e-mail, these waveforms were transferred to paper tape and mailed to California to be read into the PDP-8's core memory.

The PDP-8 was programmed to randomly present the reference and test signals in the first or second of two listening intervals per trial, interleaving the various test signals. Levels of the test sounds were adjusted according to an adaptive procedure known as Parameter Estimation by Sequential Testing (PEST). The method employed a binary search algorithm that varied step sizes and reversed directions after specifiable numbers of trials, depending on which signal of a pair the listener found more annoying. Data collection halted when a sequence of judgments for each signal pair satisfied a Wald sequential test and/or minimal step size criteria. The major findings of this study were 1) that the phase spectrum of impulsive signals did not affect their annoyance, and 2) that the annoyance of impulsive signals is appropriately modeled as an energy summation process.[i]

[i]The last psychoacoustic study in BBN's Canoga Park laboratory was conducted 32 years later. The PDP-8 was long gone by this time, replaced by a PC-based system using commercially-available rather than custom-built interfaces and signal presentation hardware. A two-alternative forced choice test protocol was used, but with a maximum likelihood ratio adaptive method rather than PEST. Ironically, the same technique used to create signals of identical power spectra but different phase spectra three decades earlier was used once again to create "fraternal twin" signal pairs. Instead of impulses, the signals in the last study were eight-second long samples that included aircraft overflights and surface vehicle pass-bys. The unprocessed signal of each pair was matched by a processed signal with identical frequency content but completely scrambled phase. Thus, the two signals were indistinguishable to a sound level meter, but readily discriminable by human observers. In the case of one of the signal pairs (a violin cadenza and its

Speech perception and production. For the National Institute of Neurological and Communicative Disorders and Stroke, Kalikow and K. Stevens developed a test to discriminate between cognitive and sensory deficits in the ability to understand speech heard in the presence of other speech — a vexing problem for many older listeners. The test involved listening for the final words of two types of sentences heard at varying levels of speech-to-noise ratios. The last words of some sentences were highly predictable from contextual cues, while the last words of others were not.

The noise used in the speech-in-noise (SPIN) test was "calibrated babble." The Los Angeles laboratory manufactured the babble by recording and then mixing the speech of multiple male and female speakers; it also provided high quality recordings of male and female speakers reciting both types of sentences. Listeners with good cognitive skills score higher on the SPIN test, at all levels of speech-to-noise ratios, for the sentences in which the initial part of the sentence is predictive of the last part than for those in which it is not.[23]

People modulate their speech in various ways in response to the situational conditions in which they are speaking. To study the details of such modulation, it is necessary to have a corpus of speech samples taken under a variety of conditions. For the Environmental Protection Agency's Research Office, BBN made extensive recordings of conversational speech levels in a wide range of communicating environments, including classrooms. One-third octave band spectra of these speech levels were analyzed by age, sex, levels of vocal effort, speaker-listener distances, and indoor and outdoor settings. Recordings were also made of the speech of many talkers under more controlled conditions in an anechoic chamber. The resulting report[24] remains a major source of practical information about speech levels under everyday conditions.

BBN also conducted studies intended to test the hypothesis that untruthful utterances are accompanied by a greater degree of vocal "microtremor" than truthful utterances. A protocol was developed in which test participants seated in the anechoic chamber of the Los Angeles laboratory attempted to persuade other test participants who could not see their faces that they were truthfully reporting the contents of a page of text in front of them — even when they were not. Each speaker's utterances were recorded and individually scored in real time by the other test participants so that it could be determined whether the degree of microtremor in the speaker's voice supported a more accurate categorization of the truthfulness of the test statements than could be obtained from the subjective judgments of the other test participants.

Obtaining informed consent for participation was one of the more difficult aspects of the study design. A set of monetary incentives for successful deception of the other test participants had to be devised that was simultaneously great enough to encourage earnest participation in the study, but not so great that an Institutional Review Board would view the incentives as coercive. If all test participants in each session were about equally effective in persuading one another of the truthfulness of their untruthful statements, the payoff to each was on the order of $50. If one test participant had markedly greater success in persuading the others of the truthfulness of untruthful statements, however, the payoff to that participant could be much greater.

Psychological response to environmental noise. In addition to conducting laboratory studies of the psychoacoustics of noise, BBNers also did many studies of the effects of, and people's responses to, noise as encountered in the environments in which

phase-scrambled twin), the differences in judged annoyance were greater than 30 dB. The study therefore established that no simple frequency-weighting network, even one intended to approximate human hearing sensitivity, can fully account for the annoyance of noise exposure. It is reported in Fidell, S., Sneddon, M, Pearsons, K., & Howe, R. (2002). Insufficiency of spectral information as a primary determinant of the annoyance of environmental sounds. *Noise Control Engineering Journal, 50*, 12-18.

they lived, worked or played. Most of these studies were done by the Psychoacoustics Department, often in collaboration with acousticians in the Los Angeles facility. Topics investigated included the effects of transformer and transmission lines on people[25] and the noticeability of a decrease in the level of aircraft noise.[26] The studies of annoyance from noise that were done in the psychoacoustics laboratory were complemented with investigations of annoyance from noise in a variety of real-world contexts, including in the vicinity of commuter aircraft overflights,[27] in the vicinity of traffic noise,[28] and in wilderness recreation areas.[29]

Although most of the studies of community reaction to environmental noise were related to effects of garden-variety air and ground traffic noise, a few studies were also conducted on more specialized noise sources, such as blast and other impulsive noises (e.g., weapons noise and sonic boom). One of these, sponsored by the Bureau of Mines, focused on the noise of strip mine and quarry blasting. Samples of residents of neighborhoods near active coal mining and quarrying sites were interviewed to relate the prevalence of annoyance in such neighborhoods with long term records of blasting activity.[30] Social surveys and laboratory studies were also conducted of reactions to corona discharge noise of 400 kv electrical transmission lines, and extensive field measurements made of low frequency masking noise for transformer tones as a function of population density.[31]

Figure 8.3. Combining psychoacoustic theory with computer technology, BBN scientists studied effects of noise on sleep.

The effect of environmental noise on sleep was the subject of several large-scale in situ studies of sleep disturbance in familiar sleeping quarters.[32] Test participants (see Figure 8.3) in these studies were given bedside buttons to push if they awoke for any reason during the night. Pushing the button produced a time-stamped list of awakenings in digital form, which was post-processed in conjunction with indoor and outdoor time series of sound pressure measurements. The processing software associated noise events occurring within specified intervals of awakening responses, as well as noise events not associated with awakening responses. BBN developed automated and efficient data collection and analysis procedures that made such large-scale studies cost-effective, beginning with a pioneering study in the early 1980s in which a PDP-8 connected by telephone lines to homes of test participants produced noise events into bedrooms in an adaptive study design.[33] This work included the development of models for predicting from measurable variables the level of sleep disturbance that noise would be likely to induce.[34]

In the early 1990s, the L.A. office conducted a set of large-scale field studies of aircraft noise-induced sleep disturbance. Indoor and outdoor aircraft noise measurements were made while test participants living near an Air Force base and three large civil airports pressed buttons if they awakened for any reason while sleeping in their own homes. Some of the test participants also wore wristwatch-like recording accelerometers ("actimeters") to measure motility as well as awakening. Published analyses of these behavioral awakening and motility findings remain the most comprehensive information about noise-induced sleep interference in residential settings.

A sonic boom simulator constructed for materials testing under a long term U.S. Air Force program was used to study the annoyance of high energy impulsive noise, and also the contribution of rattle to low frequency impulsive noise.[35] These laboratory studies complemented field surveys of low frequency aircraft noise conducted near airports in Los Angeles and Minneapolis.

High-frequency hearing. K. Stevens, Green and colleagues applied computer technology to the measurement of high frequency hearing.[36] Measurement of high-frequency hearing was notoriously problematic with conventional audiometric approaches, in part because when the ear is stimulated at high frequencies (when the wavelength of the sound is close to the length of the ear canal) by a conventional transducer, standing waves are generated in the ear canal, giving rise to resonances that make calibration difficult. In their work, Stevens, Green and colleagues explored the idea that one could estimate the sound pressure at the eardrum by first measuring the response of the ear canal to an impulse applied at its input end. From the impulse response, one could use an algorithm to estimate the transfer function from the transducer to the inner end of the ear canal. On the basis of their experimental results, they concluded that the computational approach they developed could be used to obtain audiometric thresholds for most listeners in the frequency range of 8 to 17 kHz; thresholds for higher frequencies could be obtained in some, but not all, cases. Thresholds were found to increase by about 10 dB with increasing frequency over the range of frequencies tested.

Cognition

BBN psychologists undertook many projects focused on various aspects of cognition — memory, mental models, reasoning, reading and writing, and teaching and learning. Much of the work in this category addressed the practical questions of how to facilitate the development of cognitive skills essential to education such as reading, writing, learning and teaching. Some of this work is described in the chapter on educational technology.

Memory. One of the things that attracted Allan Collins[j] to BBN was work that Ross Quillian had done in his Ph. D. program on natural language understanding by computer,

[j]When, in the mid 1960s, Swets inquired of Arthur Melton at the University of Michigan of promising candidates that might be of interest to BBN, Melton recommended Collins. Collins came to BBN in 1967 and stayed until 1989, when he reduced his commitment to BBN to half-time in order to accept a faculty position at Northwestern University. During 33 years at BBN, he was extraordinarily productive, publishing numerous journal articles and book chapters, many of which became widely cited by other researchers. He was named a BBN Principal Scientist in 1982; his impact on educational research was recognized by his election to the National Academy of Education in 1992. He was also the founding editor of *Cognitive Science.* Collins was instrumental in bringing several psychologists to BBN, including Gentner, Smith and A. Stevens, and he collaborated with all of them.

which Collins had read as a student of Walter Reitman's at Michigan. Quillian was at BBN when Collins arrived and the two collaborated on some seminal work on *semantic memory* and the idea of *spreading activation* as a process underlying memory search.[37]

Other BBN psychologists also produced empirical or theoretical work on one or another aspect of memory. Topics of reports and articles included short-term visual memory,[38] memory for sentences,[39] lexical memory,[40] long-term visual memory,[41] archival memory,[42] and inhibitory effects in memory.[43]

Mental models. A series of studies was done under the sponsorship of the Office of Naval Research on the topic of mental models. A mental model is a mental representation of some aspect — object, process, relationship — of the real world used to support explanation and prediction, often by mental simulation. Discovery of how people represent things mentally is important to education; failure of a mental model to correspond to the reality it is supposed to represent can lead to various types of difficulties, so an important goal of education is to try to ensure that the models students acquire are not defective in significant ways.

A specific goal of the mental models research done at BBN was to characterize how people could simulate in their mind's eye how different systems behave. The work supported the development of computer tutors that could help learners construct mental models of complex systems. The work on this topic was performed primarily by Collins, Gentner, Smith and A. Stevens, and is documented in numerous publications.[44] The concept of mental models also was applied in the work on educational technology, especially by John Frederiksen and Barbara White.[45]

Reasoning. Collins worked with a number of colleagues on several studies of human reasoning with support from the Advanced Research Projects Agency, the Army Research Institute, and the Office of Naval Research.[46] With the sponsorship of the Office of Naval Research, Gentner did a series of studies on analogical reasoning. She developed a theoretical account of analogy as structure-mapping[47] and applied it in several contexts.[48] The 1983 theoretical account was chosen as one of 10 Classic Articles by *Cognitive Science*. Work on reasoning was also done by other BBNers.[49]

Reading and writing. Much, though not all, of the work of BBNers on reading and writing was done under the University of Illinois-BBN Center for the Study of Reading. This work drew from a variety of project areas at BBN — cognitive psychology, speech and natural language processing, and educational technology. It addressed many aspects of the problem of becoming a competent, comprehending reader for purposes of learning and pleasure. Analyses of materials used for teaching reading, for assessing text readability and for testing reading competence revealed many ways in which reading instruction could be improved. Communication of the results of these studies to educators and to text-book publishers was an objective of the Reading Center over the entire course of its existence, and was realized through numerous publications, conferences, symposia and informal interactions.

The principal investigator for BBN's part of the Reading Center was Bertram Bruce. Other BBN contributors to the work of the center included Adams, John Seely Brown, Collins, Frederiksen, Gentner, Huggins, Kathie Larkin, Ann Rosebery, Andee Rubin, Beth Warren and Bonnie Weber. A list of the numerous reports issued by the center over the duration of its existence is posted at `http://www.ed.uiuc.edu/BER/csr/Tech-rep.html`. The range of subjects relating to reading, within the Reading Center and other projects, was broad.[50]

Notable among the products of BBNers on reading was the book, *Beginning to Read: Thinking and Learning about Print*, by Adams (published by the MIT Press in 1989),

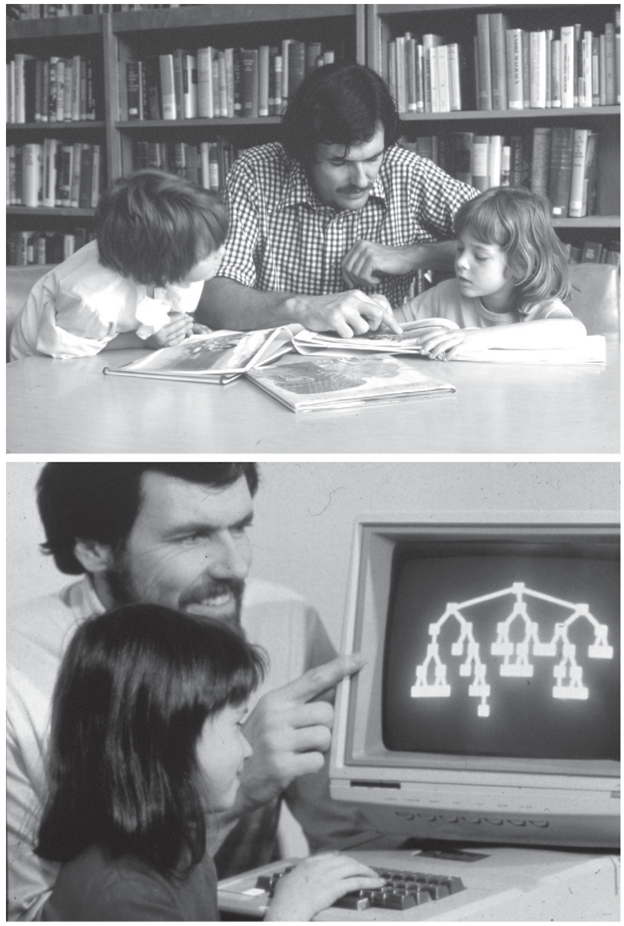


Figure 8.4. Top: Bertram (Chip) Bruce with participants in reading/writing projects. Bottom: The child is constructing a story with the help of QUILL.

which received national acclaim and for which the author was given the Sylvia Scribner Award from the American Educational Research Association. The Scribner Award is given annually to recognize work that has had extraordinary impact on educational research during the preceding 10 years. This book was featured by several reviews and commentary in an issue of *Psychological Science*, and received considerable attention in a feature on reading in *Time Magazine*.

The study of reading, and especially the problem of learning to read, naturally leads to questions relating to writing and the problem of learning to write. It is not surprising to find, among the reports produced by the Reading Center work, a number that address one or another aspect of these topics. [51] A three-year project explicitly addressed to writing was sponsored by the Center for Libraries and Education of the U.S. Department of Education. The main purpose of this project, which was directed by Bruce and Rubin, was to facilitate the learning of writing skills by creating "a classroom version of the powerful writing environments we used for our own writing." (See Figure 8.4.) Reasons for expecting this approach to be effective are spelled out in several publications.[52] A major result of this project was the development, testing, and distribution of QUILL, which provided classroom access to the types of writing tools envisioned by Bruce and Rubin, tailored for use by grade-school students. A full account of the project is given in Bruce and Rubin's 1993 book, *Electronic Quills: A situated evaluation of using computers for writing in classrooms*, published by Erlbaum.

A project sponsored by the National Institute for Child Health and Human Development involved the development of a test of decoding skills that could be used with children at the age when they are beginning to learn to read. The test was to assess children's ability to translate from the orthography of printed text to the phonetic representation of speech and to identify specific problems that some children experience in this regard.[53]

Work was done for the Office of Naval Research on the question of what determines the comprehensibility of written instructions. Smith and his colleagues investigated the effectiveness of explanatory material for instruction comprehension and the importance of individual differences in instruction understanding.[54]

Teaching and learning. Several projects on teaching — identifying tutorial strategies that teachers use, the building of intelligent tutoring systems — were sponsored by the Office of Naval Research. Much of this work — involving Scholar and the WHY system, among others — is described in the chapter on educational technology. Work aimed at identifying tutorial strategies that teachers use grew out of earlier work on the WHY system and was done by Collins and A. Stevens.[55]

An influential idea regarding teaching and learning championed by Collins, Brown and Susan Newman was that of cognitive apprenticeship as an effective means of learning.[56] This work has been widely cited in the educational research literature and has had considerable influence on thinking about classroom practice. The theme of cognitive apprenticeship, and the closely associated one of situated learning[57] were pursued in additional Reading Center projects as well as in some work sponsored by the Office of Naval Research.

The National Institute of Education sponsored not only the Center for the Study of Reading but other contracted work at BBN as well. One such contract, which was won by a competitive bid and yielded a number of reports, [58] was for a study of the teaching of higher-order cognitive skills. Another educational project, this one sponsored by the Educational Technology Center at the Harvard Graduate School of Education, involved the organizing of a conference to discuss and reflect upon how technology could, or should, affect education over the following few decades [59]

Project Intelligence. Among the more unusual projects undertaken by BBN psychologists was "Project Intelligence," which started with a request from Luis Alberto Machado, Minister of State for the Development of Human Intelligence — a then newly created ministry in Venezuela — to Harvard University, via José Buscaglia,[k] to undertake a project to, in his terms, increase the intelligence of children in Venezuela. Minister Machado was a firm believer that intelligence was determined, to a large extent, by experience, especially by events in early childhood. A visionary and activist, he had aggressively promoted the idea that the state has an obligation to see that every child has the opportunity to develop his or her potential intelligence to the fullest. The kind

[k]José Buscaglia is a sculptor, well-known internationally especially for his numerous public monuments and sculptural groups in the United States, Peurto Rico, Spain and the Virgin Islands. BBNers from the period will remember his marvelous, larger-than-life, stone statue of Robert Frost that graced the BBN courtyard for several years, and stands today on the grounds of Merrimack College in Andover, Massachusetts. Buscaglia, who knew Minister Machado, had carried the minister's invitation to Harvard to undertake an educational project in Venezuela. Later, when BBN became involved, Buscaglia joined the BBN staff for the duration of the project, before returning to full-time work as a sculptor. He was not only invaluable for this particular project, but a delightful colleague in every way — his energy and creativity seemed boundless and his sense of humor a constant plus.

of project that Minister Machado had in mind was not something that a university could easily undertake. Richard Herrnstein, the Harvard professor on whose desk the request from Venezuela finally landed, approached us at BBN to see if we might be interested in attempting to define a project on which Harvard and BBN could collaborate. Eventually a project was defined, the goal of which was to develop and test an experimental course with the objective of improving the thinking skills of seventh-grade students in selected schools in Venezuela.[l]

The project, which ran about four years, was funded by Petroleos of Venezuela.[m] The principal investigator for Harvard was Professor Jorge Dominguez, and for BBN, Nickerson. Senior advisors were Herrnstein of Harvard and Swets of BBN; the project director for the Ministry of Education, Republic of Venezuela, was Margarita de Sanchez. Other major contributors to this project included Adams, Buscaglia, Feehrer, Getty, Grignetti, Susan Herrnstein, Huggins, Catalina Laserna, David Perkins, and Brenda Starr. The results were described in a final report delivered to the government of Venezuela in October, 1983[60] and, in part, in several subsequent publications.[61] Following completion of the Venezuelan project an English version of much of the curriculum, suitable for use in the United States, was published under the title *Odyssey* by Mastery Education in 1986.[62]

Driving and highway safety

Attentional demands of automobile driving. John Senders and mechanical engineer Charles Dietrich collaborated in the 1960s on some studies of the attentional demands of automobile driving. They instrumented a crash helmet with a visor that could be programmed to fall and rise on a predetermined schedule. In its lowered position, it occluded the wearer's vision. By varying the up-down schedule of the visor — and thus the frequency and duration of the wearer's glimpses and occlusions — the investigators could determine how much visual information a driver required to maintain control of a vehicle and how this depended on driving conditions. This work led directly to the project next discussed. (This research was done before the days of Institutional Review Boards that now have to approve all government-sponsored experiments involving human subjects and ensure that they comply with government safety standards; there is some question as to whether it would now be possible to get approval for experiments requiring people to drive an automobile while wearing a helmet that permitted them

[l]Initially, four of us — Buscaglia, Herrnstein, Nickerson and Swets — went to Caracas to learn more about what was desired and whether it made sense for us to get involved. After a couple of days of exploring, we decided that it did not. We were uncomfortable with the language of "raising intelligence," nervous about the political exposure of Minister Machado's office and activities — the press had not been overly sympathetic to his aspirations — and fearful that unrealistic expectations might have been promoted regarding what could be achieved. Shortly before our scheduled departure, we informed Minister Machado of our decision. He seemed not to be greatly surprised, and asked us to share with him our thinking, which we did. His reaction was that he understood perfectly — but what kind of project would we be willing to undertake and under what conditions? Eventually, after considerable deliberation, we proposed to design an experimental course, to help Venezuelan teachers give it, and to report the results, whatever they were, in the open literature. The proposal was accepted without reservation, and that is what we did.

[m]Petroleos of Venezuela was the largest company in the country. We sometimes found ourselves giving progress reports in the boardroom of this corporation to the members of the board. This group, which was presided over by a retired general was always cordial and supportive, but the setting was formal and somewhat imposing, nonetheless. The general obviously commanded great respect and deference. We got to see him from a different perspective when he appeared, unaccompanied and with no warning, one day at BBN in Cambridge, dressed casually and in tennis shoes, to pay an informal visit. It was a relaxed and pleasant day; the general seemed to enjoy the visit immensely, learned a bit about a number of ongoing projects, had a casual and chat-filled lunch with a miscellany of BBNers, charmed us all, and disappeared at the end of the day as quietly and unceremoniously as he had arrived.

visual input only some fraction of the time, even in a dual-controlled automobile with a back-up driver.[n])

Vehicle rear lighting. A large percentage of highway accidents involve lead vehicle's rear-end collisions. The purpose of a project sponsored by the U.S. Department of Transportation was to investigate the effectiveness of some innovative rear-lighting systems in providing information to a following driver regarding the behavior or intentions of the driver of a leading vehicle. The work required a team composed of engineers, psychologists and applied mathematicians. We instrumented a vehicle with an experimental rear-lighting system that could represent in analog fashion the vehicle's speed or acceleration/deceleration, as well as with radar and computing equipment that would permit a continuous record of the vehicle's speed and the distance between this vehicle and a following one. With this system, we did car-following experiments on a stretch of interstate highway that was under construction and had not yet been opened to traffic. The task of the driver of the following car was to maintain a constant distance behind the lead car under a variety of scheduled maneuvers by the lead driver and with different operating characteristics of the lead vehicle's rear-lighting system. In addition to collecting empirical data, we developed a mathematical model of the driving task. The final report[63] contained several recommendations, the first of which had to do with the perceptual separation of brake lights from running and directional lights:

- Insofar as possible, the principle of perceptual redundancy should be used in the encoding of messages to be conveyed via the rear-light system.
- In particular, at the very least, brake lights should be distinct from running and directional lights with respect to at least two perceptual dimensions. Moreover, the differences along each dimension should be sufficiently great so as to minimize confusions.
- What the coding dimensions should be is an issue of somewhat lesser importance. Our recommendation, however, is that position be one and that color be another. Specifically, the brake lights should be in a different position, and a different color, than either running or directional lights.
- By different position we intend that there should exist a clear and distinct boundary such that if only one signal is illuminated an observer should be able to identify which signal it is (p. 198).

Whether this recommendation was a factor in the government's later decision to mandate that brake lights, spatially separated from running lights, be located at the level of the rear window, we do not know. We like to believe, of course, that it was.

The report contained a number of other recommendations, ranging from some that we believed could be substantiated by available data to others that we believed had a compelling rational basis to still others that we described as tentative and requiring further investigation.

Social survey research

Surveys have been a mainstay of BBN business since its founding, especially surveys of noise — aircraft noise, road vehicle noise, industrial noise — in the vicinity of airports, highways and in residential areas. These surveys typically involved making noise

[n]A video, "Pioneer days on Rt 128" demonstrating the use of the helmet can be seen at `http://www.youtube.com/watch?v=kOguslSPpqo`.

measurements, often for the purpose of determining compliance with EPA regulations, or to inform efforts by BBN scientists and engineers to develop noise abatement devices (more effective mufflers) or techniques. A review of that work, which was extensive, is far beyond the scope of this chapter. Here we wish only to note some social survey work, typically involving the collection of data, sometimes in the form of responses to written questionnaires or interviews, done by BBN psychologists. Sometimes these surveys involved people's reactions to noise in their communities or their work or recreation environments; sometimes they involved entirely different types of issues. A key figure in the social survey work at BBN was Glenn Jones,[o] who came to BBN in 1967 from the University of Michigan's Institute for Survey Research, and served as BBN's resident expert in this area for the next 10 years.

Human response to noise. Not surprisingly, given BBN's history of work in acoustics, several surveys involved human (individual or community) response to noise. Survey studies were done for the Motor Vehicle Manufacturer's Association[64] and the U.S. Department of Transportation[65] on response to vehicle noise. Human response to sonic booms was also a topic of investigation.[66] A survey study at the Los Angeles International Airport documented the ineffectiveness of a late-night curfew.[67]

Jones was indirectly responsible for a series of community noise surveys conducted at BBN after his departure, including the first nationwide urban noise survey and a range of studies on community response to aircraft noise conducted at large and small airports in the United States and Canada.[68] Studies of these types contributed to a database of findings that Theodore Schultz, another BBNer, analyzed in a highly influential meta-analysis of social survey findings.[69]

Safety and use of consumer products and other topics. Some of the surveys done by Jones and colleagues pertained to issues of safety and use of consumer products. One study, for the Juvenile Products Manufacturers Association focused on the use of children's car seats and related devices.[70] Another for the same association investigated mothers' experiences with cribs and other children's furniture.[71] Survey or questionnaire techniques were also sometimes used in studies aimed at providing a better understanding of the effects of specific variables in operational situations.[72] A survey for NASA explored the opinions of 132 commercial airline pilots regarding issues of automation in aviation contexts.[73]

Enhancing human performance with visual displays

Medical imaging. Work on medical imaging began under the leadership of Swets in the mid 1970s. This work sustained a long collaboration between Swets and colleagues Getty (Figure 8.5) and Pickett, among others, and sponsorship by several agencies.[p] It included evaluation of non-ionizing imaging modalities, development of computer-based instructional programs, optimization of utilization of imaging tests, development of a system for stereoscopic digital mammography, and development of techniques for

[o] Jones was hired by Swets on the recommendation of William McKeachie, then chair of the University of Michigan's psychology department. Sam Labate, BBN's CEO at the time had some interest in the possibility of acquiring an existing social survey organization as a way of establishing BBN in a new area of activity, but such an acquisition did not materialize. Jones was somewhat older than the average BBN psychologist when he joined BBN; he was extremely well-liked and became viewed as a very personable and unassuming, but meticulous, elder statesman.

[p] The National Cancer Institute, the National Library of Medicine, the National Center for Health Services Research, the National Eye Institute, the Agency of Health Care Policy Research, and the U.S. Army Medical R&D Command.

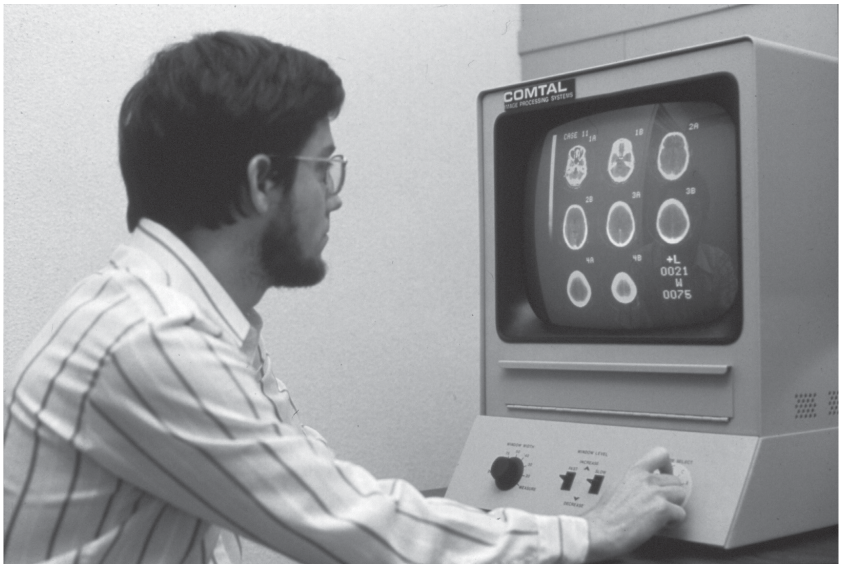


Figure 8.5 David Getty working on medical Imaging problems.

enhancing the radiologist's accuracy in interpreting image-based studies (e.g., mammograms, CT studies of the liver, MRI studies of the prostate).

Initially the focus of the work was on evaluation, comparing the effectiveness of different imaging techniques. The objective of the first project, sponsored by the National Cancer Institute, was to develop a standard protocol for the evaluation of imaging techniques in cancer diagnosis. The new protocol was applied to three alternative forms of mammography and to the then-new computed-tomography scanner.[74] The primary focus of the work changed over time to enhancement, the objective becoming that of finding ways to increase the accuracy with which images could be read for diagnostic purposes. Image inspection techniques and decision-making aids designed to increase the accuracy of image-based diagnoses were developed and tested. The techniques that were developed complemented a human-factors approach to improve the radiologist's perceptual judgment capabilities with a computer-assisted diagnosis; they provided a 15 percent gain in either sensitivity (the probability of correctly calling a malignant lesion malignant) or specificity (the probability of correctly calling a benign lesion benign) of diagnosis in laboratory studies.[75] In one test, the diagnostic performance of community radiologists reading mammograms was raised to the level typically obtained by specialists.[76] Later studies refined the methods of aiding the reading and diagnosis of mammograms[77] and for combining evidence from multiple imaging modalities.[78] Positive results were also obtained with techniques applied to the interpretation of images for purposes of diagnosing cataracts[79] or liver lesions[80] and the staging of prostate cancer.[81] This work was supported by the National Cancer Institute and the National Eye Institute.

The work on medical imaging nicely illustrates how projects conducted for one purpose could produce results that proved to be useful in projects motivated by very different interests. The results of studies of recognition, identification or classification of complex visual patterns, conducted for the U.S. Office of Naval Research[82] were used to good effect in the development of image interpretation techniques. In particular, the application of multidimensional scaling procedures in the ONR-sponsored work to reveal perceptual features led to the formulation of mathematical models that predict the rates of identification confusion error for complex visual patterns based on the observer's multidimensional perceptual space. This approach was effectively applied to the enhancement of medical imaging interpretation.[83]

Recent work is perhaps best characterized by describing a general system to which it points. In such a system, speech understanding technology will be coupled with other advancements in the design of a system that will recognize the image reader's spoken scale values, generate a standard prose report for the referring physician or surgeon, automatically tailor a computer-based instruction program to his/her needs, be available for reference in reconciling multiple opinions via networked multi-media video conferencing, and for entering into a computer-based prediction rule along with quantified features from other imaging modalities and clinical examinations for the best overall, single decision. The diagnostic probabilities will be available to help calibrate treatment recommendations within and across individual radiologists in accordance with agreed upon decision rules.

Perceptual properties of volumetric displays. In the 1980s, Getty and Huggins conducted studies for the Office of Naval Research on perceptual properties of true volumetric displays, using Larry Sher's SpaceGraph display.[84] At the request of the Office of Naval Research, Getty organized a conference on 3-D displays held at the National Academy of Sciences.[85] He remained interested in stereoscopic vision and decided to try to apply stereoscopic imaging to mammography as a way of improving early detection of subtle lesions in the breast and reducing false positives. He developed a stereoscopic capture and display system that permits a radiologist to view the internal structure of the breast in depth, and was awarded a patent for the system, generalized to "stereo radiography." With BBN IR&D funding, Getty and Huggins developed a prototype high-resolution stereo display workstation in 1992, that permits either manual or speech control, and modified a research digital mammography unit at the University of Massachusetts Medical Center (UMMC) to acquire stereo mammograms. (A stereo mammogram consists of two x-ray images of the breast taken from slightly different points of view; when the radiologist looks at the images on a computer-based display, he/she sees the breast in depth.) Funding was obtained from the Army's Breast Cancer Research Program in 1996 to conduct a study on patients at UMMC of the value of stereoscopic digital mammography as an adjunct to standard film mammography for diagnosing breast cancer. Stereo imaging significantly improved diagnostic accuracy, and there was suggestive evidence that mammographers were able to detect subtle lesions in stereo mammograms that were not visible in corresponding standard film mammograms.[86] Getty developed what we believe was the first multi-image format for digital medical images.[q] At the time of the writing of this chapter, Getty, who came to BBN from Brown University in 1976, continues work on the development of stereo imaging techniques to improve the detection of cancer.

Speech training. The focus of two BBN projects was the application of computer technology to the teaching of speech. The first of these, sponsored by ARPA, involved the development around a PDP-8 computer of a system to use visual displays of specific aspects of speech sounds to assist a learner of a new language to master the pronunciation of words in that language.[87] This project is described in Swets's and Feurzeig's chapters in this volume.

[q]EMI allowed the radiologist to view at one time only a single CT slice that filled the screen. Because the images were fewer than 200 pixels across, the displayed pixels were huge. To reduce perceived pixel edge artifacts, the radiologists would move back 3 or 4 feet from the display. Getty realized that the granularity problem could be solved by reducing the image size and that with smaller images several of them could be displayed at once. As an added benefit of this format, the radiologist could now compare adjacent slices.

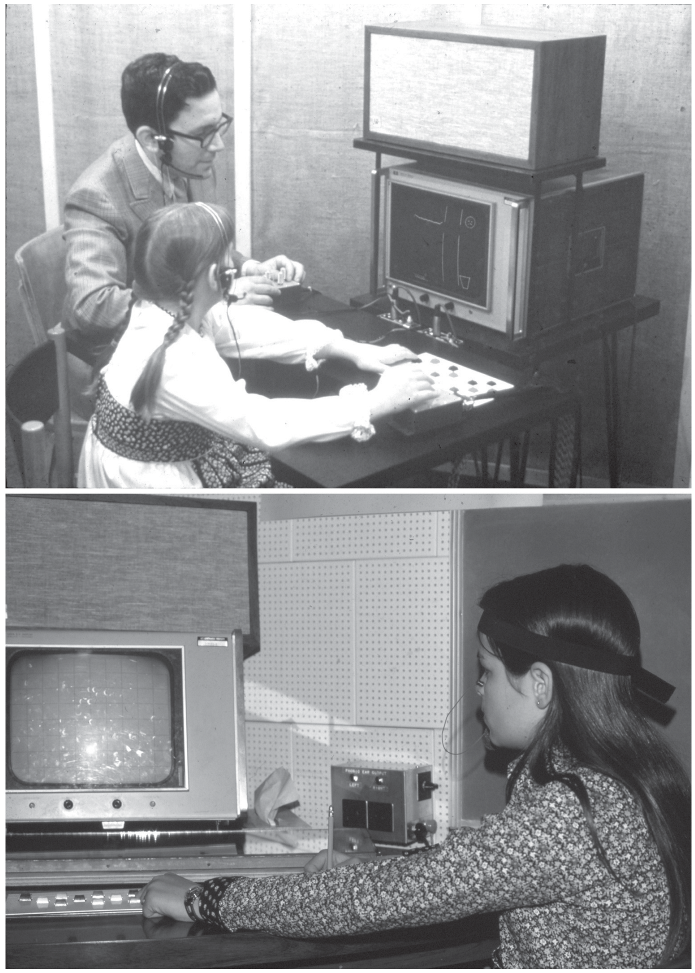


Figure 8.6. Top: Dan Kalikow working with a hearing-impaired student on the BBN speech-learning-aid system; see Figure 13.30 on page 328 for a picture of one of the displays developed for this system. Bottom: Hearing impaired student working with the same system, but a different program.

The second of these projects was an effort to develop computer-based visual displays that could be used to help teach profoundly hearing-impaired children to improve the intelligibility and quality of their speech. This project, sponsored by the Bureau of Education for the Handicapped, built on the second-language-learning work. Several visual displays were developed — again using a PDP-8 — to provide visual representations of various aspects of speech (volume, pitch, nasality, timing) that were believed to be important determinants of intelligibility and that deaf children have difficulty learning to control (see Figure 8.6). This project is also described in Feurzeig's chapter.[88]

If one of the authors may be permitted a personal word, I (Nickerson) found this project to be, at once, one of the more rewarding and more frustrating on which I had the opportunity to work at BBN — rewarding because of its objective, frustrating because of how little progress we were able to make toward realizing it. Acquisition of the ability to speak intelligibly is an extraordinarily difficult challenge for a prelingually deaf child.[89] I believe our project made some modest headway on the problem, thanks in large measure to the superb cooperation we received from the Clarke School for the

Deaf in Northampton, MA, and the Lexington School for the Deaf in New York City, where the system was used experimentally, under the direction of Dr. Arthur Boothroyd in Northampton and Ms. Janet Head in New York. We remain hopeful that lessons learned in this project can inform other attempts to bring computer technology to bear on the problem, especially as computer technology has progressed to the point that it is possible to package very large amounts of processing capability in wearable devices.

Facilitating non-oral communication by people with hearing impairment. In addition to the project just mentioned, there were other BBN projects the objective of which was to apply computer technology to the facilitation of communication by people with profound hearing impairment. This appeared also to be a problem for which the technology should have something useful to offer.[90] One project, called the Vidvox project and sponsored by The Sensory Aids Foundation, was to test the feasibility of a speech-reading aid for deaf people, based on accounts of an English member of parliament who was able to keep up with parliamentary proceedings in real time, with the aid of a computer program that translated the output of a stenographer into phonetic text. The objectives of the project were to determine if the BBN speech recognizer could produce an appropriate string of phonemes in real time[91] and whether deaf students could learn to read the output.[92] The human factors part of the study started with accurate phonetics transcriptions, and, unsurprisingly, found that students could quickly learn to read them rapidly. Unfortunately, when errors were introduced into the transcriptions, performance quickly fell off even at rates much lower than the transcription readers were capable of achieving.

Another study attempted to aid deaf children in their normal school work, especially their writing (composition), by building interactive computer games and activities that would develop, for example, their control of syntax.[93] One unanticipated finding was the delight with which the deaf students took to email, which was provided in the network's computers as an afterthought. It gave them their first opportunity to communicate *privately* with an individual friend; sign language (which all of them spoke) is "broadcast", and can be read from across the room by anyone who happens to be looking. Email provided a clear and immediate purpose for improving one's expressive writing skills.

Human-computer interaction

In view of the great influence that Licklider had in getting BBN into computer technology and his own keen interest in person-computer interaction, especially as expounded in his "Man-computer symbiosis" classic,[94] it is not surprising that interest in this subject continued to be strong among BBN psychologists long after he left. That interest was enhanced by the fact that we used computers daily for a variety of purposes, initially to control experiments, later — when desktop terminals and word-processing software became commonplace — to write papers and to communicate with colleagues. A few publications reflected ideas about human-computer interaction, often gained from these experiences.[95]

Somewhat paradoxically, perhaps, there were no funded projects with the explicit title of human-computer interaction, but there were many that related directly to the design of specific computer-based systems that were intended to be used by people in an interactive way, or whose purpose was to facilitate communication among people. Some of the projects that best illustrate this work are described in what follows.

Dialog specification and interface design. In 1975, BBN contracted with the U.S. Department of Agriculture to develop a "dialog specification procedure" that could be used by its programmers, who had previously worked only on batch processing applications, to develop software user interfaces for workstations in county field offices. It was important that the interfaces developed by different programmers have a common look and feel. Pew was the principal investigator and Rollins a major contributor. In addition to articulating general principles of good interface design, providing examples of task analysis, and specifying how to document the sequence of screens that would form the dialog, Pew and Rollins produced a detailed style guide, which included layout sheets that could be used to specify the way the screens should look and to make it easier for programmers to generate the required code.[96]

Teleconferencing. Teleconferencing work was done by BBNers in collaboration with MIT's Lincoln Laboratory. The objective of the research was to compare the relative efficacies of various protocols for management of secure, voice-only teleconferences characterized by significant variation in the bandwidth and quality of communication resources available to participants. The BBN effort was led by Feehrer, with major contributions being made by Paul Weene. D. Miller and Pew. The nature of the conferencing hardware and software required BBN to formulate a unique set of scenarios and experimental methods that required each participant to attempt to make timely inputs in order to aid in the joint solution of problems posed to conferees. The resulting scenarios and methods proved capable of exposing the weaknesses, as well as the strengths, of conferencing systems being considered for deployment.[97]

Information management, automation and workload assessment. BBN psychologists worked on a variety of projects that we find convenient to group under the rather broad umbrella of information management, automation and workload assessment. All of these projects had to do with people interacting with computers or other artifacts of information technology as a major aspect of their jobs.

A project dealing with automation in the airplane cockpit is illustrative of work in this area.[98] In 1989, BBN won a five-year task order agreement with NASA Langley, which helped solidify BBN's role as a significant player in the field of cockpit information management and automation. Tenney, Pew, Rogers and Salter assisted in the experimental evaluation of Faultfinder, a prototype cockpit fault management expert system that Eva Hudlicka (later a BBNer) had helped to develop.[99] Related work included an analysis of the application of AI to the aiding of flight crews in the performance of their tasks,[100] a report on human error in advanced maintenance control centers,[101] and a study of the automating of maintenance instructions.[102]

Getty, Swets, Pickett and David Gonthier conducted laboratory experiments supporting analysis of detection performance and sensitivity of cockpit decision aids such as windshear or collision alerts. The major contribution was to describe and identify the importance of the positive predictive value (PPV) of an alert. The PPV is the probability, given that an alert has occurred, that it was not a false alarm. This is to be distinguished from the more familiar hit rate, which is the probability that an alert will activate, given that the threatening condition of interest has occurred. Their analysis, which was derived from signal detection theory, as elucidated for the aviation context in a tutorial written for the project,[103] was published in the inaugural issue of the *Journal of Experimental Psychology: Applied*.[104] In closely related work, Swets and Getty wrote a report for NASA describing research to identify sensitivity and threshold requirements for human-centered decision aides for aircraft cockpit applications.[105]

For the Air Force Armstrong Laboratory's Crew System Ergonomics Information Analysis Center (CSERIAC), Adams, Tenney and Pew prepared a monograph concerning the psychological and human engineering literature on human attention, perception, memory, cognition, and decision-making as it pertains to the unique workload demands associated with goal-directed activities and situational awareness in complex, semi-automated work environments, such as air traffic control.[106] Addressed to engineers and designers, the goal of the report was to develop a conceptual framework that structures the problem area so as to highlight the relevance of this work to issues of system design and training.

Adams, Deutsch, Huggins, Pew, William Rogers and Tenney conducted an extensive literature review of the concept of situation awareness They developed a theory of the process by which situation awareness is acquired, reviewed existing measurement methods and suggested some novel approaches to measurement.[107] Pew and Getty also prepared a four-day course on the design and conduct of aviation simulation experiments. The course was given at NASA/Langley to a group of simulation and human factors engineers and subsequently prepared in the form of annotated slides so that it was available as a self-contained tutorial.

Modeling

Human operator modeling. Closely related to the topics of information management, automation and workload assessment is that of human operator modeling. BBN has a long history of work in this area. Before coming to BBN, Elkind had begun working on models to describe manual tracking performance,[108] and had discussed the topic with Licklider while both were at MIT. Licklider describes some of Elkind's work and credits him as the origin of "many of the ideas and many of the results described" in his chapter[109] on "Quasi-linear operator models in the study of manual tracking," in a book on mathematical psychology edited by Duncan Luce. Interest in models of manual control was maintained for several decades at BBN, largely due to the influence of Baron and William Levison.

Projects involving control theory and its applications by BBNers are described in a chapter by Baron in this volume. Here we only mention a few to give a sense of the way in which the concept of operator modeling expanded over the years to include not only supervisory control, but the modeling of human information processing and performance more generally. Modeling of supervisory control was applied to nuclear power plant operation for the Oak Ridge National Laboratory;[110] modeling of human information processing was done for DARPA in the interest of identifying ways in which performance could be enhanced by computer aids;[111] critical reviews of modeling as applied to human-machine systems and simulations of same were prepared for the U.S. Air Force;[112] the modeling of flight crew procedures in approach and landing was done for NASA;[113] and work on the modeling of human error in the D-OMAR system was performed for the same agency.[114]

Source- and observer-based aircraft noise contouring. Work done in aircraft noise contouring illustrates nicely how a project could draw on BBN's expertise in acoustics, psychology and computer technology, so we describe it in some detail. The introduction of jet transports into U.S. domestic service in 1958 exposed populations living near large civil airports to unprecedented levels of aircraft noise, and civil aviation to unprecedented levels of political and legal challenge. Under contract to the U.S. Air Force in the late 1960s and 1970s, staff of BBN's Los Angeles office (principally William Galloway, Dwight Bishop, Horonjeff, Nicholaas Reddingius, and Rao Kandukuri) developed NOISEMAP, the first systematic aircraft noise contouring software, to quantify noise exposure produced by landings and takeoffs near airport runways.

The initial use of NOISEMAP was to prepare source-based noise emission contours for the vicinity of military airfields. Working from a database of aircraft noise measurements and empirical noise-power-distance curves derived from aircraft noise measurements made by BBN since the 1950s, the software deterministically modeled cumulative noise exposure produced by propagating noise isotropically from moving point sources to ground locations of interest.

Written originally in Fortran, NOISEMAP ran in batch mode and produced a grid of noise exposure values through which contours were interpolated by hand. As the software matured, it incorporated many convenience-of-use features and enhancements to its capability and generality of application. By the mid-1970s, the U.S. Air Force and Navy were routinely using NOISEMAP to prepare noise exposure contours at scores of air fields, and in a variety of noise metrics. NATO members and Australia also adapted NOISEMAP to their own purposes. By the late 1970s, FAA began development of its own Integrated Noise Model (INM) software. Many years and millions of software development dollars later, much-rewritten and highly evolved INM software has grown into a very capable aircraft noise modeling system. NOISEMAP remains the software of choice for modeling noise from military aviation.

Under the Air Force's Noise and Sonic Boom Impact Technology Program in the late 1980s, BBN pioneered the linkage between aircraft noise modeling software and geo-information system (GIS) software. A PC-based "Assessment System for Aircraft Noise" (ASAN) was created to permit Air Force environmental planners to identify sensitive land uses underlying airspace reserved for Military Training Routes and Military Operations Areas, and to describe aircraft noise impacts in very large areas remote from military bases. Passage of Public Law 100-91, the National Parks Overflight Act of 1987, created a need for a form of aircraft noise modeling different from the then conventional source-based modeling. Most of the work in the ASAN system was done by the psychoacoustics group in the Los Angeles office, but some work on interface design and training materials was done by Papazian and Tenney in Cambridge.

Source-based modeling of noise emissions answers the question "How much noise does an airplane flying *here* create *there*?" To protect natural quiet from aircraft noise intrusions in park and wilderness settings remote from airports, however, the relevant question is "From what volume of airspace must aircraft be excluded to prevent their noise from being heard within a specified area?" For the latter purpose, the more relevant form of noise modeling is observer-based rather than source-based. BBN (primarily Fidell, Michael Harris, Reddingius, and John Smythe) therefore created the National Park Service's Overflight Decision Support System (NODSS), which included novel, GIS-based software capable of producing observer-based audibility contours. NODSS explicitly considers the spectral characteristics of noise produced by aircraft, and the (frequency-dependent) effects of atmospheric absorption and barrier diffraction. Indigenous noise levels at an observer's location are also considered in calculation of integrated (duration-weighted) audibility of low-level noise intrusions. The program further calculates a variety of audibility-based metrics to facilitate the interpretation of aircraft noise intrusions in units of minutes of noticeable noise intrusions. Passage of Public Law 100-91, which declared "natural quiet" to be an important resource in park and wilderness areas managed for outdoor recreation, led to the hiring of BBN by the U.S. Forest Service and the National Park Service to measure indigenous and aircraft noise levels at remote sites on public lands.[r]

[r]In one such measurement exercise, the sections of a 10 meter high meteorological tower proved too long to load onto pack animals negotiating tight switchbacks in the Golden Trout Wilderness of California's Sierra Nevada. Pearsons, Ron Mucci, and Fidell thus joined the mule train as porters, carrying the acoustic instrumentation, cables, computers, batteries, and camping equipment into the wilderness. After the pack

Probabilistic noise exposure and complaint modeling. Although source-based and observer-based approaches might seem to exhaust all of the reasonable perspectives on aircraft noise modeling, yet another approach was developed in BBN's Los Angeles office in the late 1990s. Both source-based and observer-based noise modeling are place-oriented and deterministic, in the sense that users specify operational characteristics such as flight paths, times of day, and numbers of aircraft operations. The third perspective is a joint probability approach to estimating personal (as distinct from place-oriented) noise exposure. The joint probability approach is most appropriate in circumstances of sporadic exposure to aircraft noise in non-residential settings.

In vast areas of public lands underlying military flight operations areas, aircraft noise intrusions are unscheduled, unpredictable, and rare — but occasionally of very high short-term level. Likewise, outdoor recreational uses of such lands (hiking, camping, picnicking, etc.) are episodic, and people occasionally exposed to aircraft noise are not always located at fixed and predictable locations. In such circumstances, the primary concern may be *when* rather than *where* the noise sources and receivers are found with respect to one another. Two parties of hikers, for example, may leave the same trailhead on the same itinerary at different times of day. One may encounter no aircraft noise whatsoever during its visit, while the other may be overflown at low altitude and high speed by one or several aircraft.

BBN began development of prototype software known as RECMAP to yield predictions of likelihoods that aircraft and visitors come into sufficient proximity to one another over a given time period to experience personal noise exposure. Rather than cumulating the noise exposure produced by a pre-determined set of aircraft operations at all points within a grid of fixed points, RECMAP uses iterative Monte Carlo techniques to run simulations of interactions between airborne and groundborne moving sources. Thousands of simulations may be conducted in which scenarios involving varying numbers and types of aircraft operations and visitor activities within given airspace boundaries and land areas are evaluated. Distributions of noise levels are generated for each iteration, and described statistically to yield expectations of a range of exposure statistics for the experiences of individual visitors. The resulting information is of greater utility for environmental disclosure purposes than simple descriptions of long term, area-wide average exposure values.

In the late 1990s, BBN staff (primarily Fidell, Sneddon, and Howe) developed methods for directly contouring noise effects rather than noise exposure. Airport noise monitoring systems (based on automated digital noise monitoring hardware pioneered by BBN in the early 1970s) had evolved by the early 1990s to the point that they began to accumulate organized files of time-tagged aircraft noise complaints. Fidell and co-workers geo-referenced (assigned latitude/longitude coordinates) to the street addresses of complainants, and then used off-the-shelf GIS software to produce representations of complaint densities as false elevation in pseudo-terrain. On such maps (as, for example, in Figure 8.7), elevation is proportional to complaints per square mile per month. The resulting graphics reveal information about the geographic extent

train had departed and the tower had been erected in the middle of an 8 element acoustic array with a 500 meter aperture, it was discovered that the password for a critical program had been left at the office. Pearsons, a committed jogger, ran 14 miles at an altitude of 7500 feet in steep terrain to the nearest telephone to retrieve the password in time to start the monitoring work on schedule. Other anecdotes from similar studies include programming by kerosene lantern with ants crawling on the keyboard, range cattle defecating with unerring aim on microphone cables running through the woods, and bears reminding researchers of their relative positions on the food chain. Still others concern jumping out of helicopters into snow to defend tranquilized deer equipped with tracking collars from wolves. (The researchers figured that it would be better to seek forgiveness than permission should the need arise.)

of aircraft noise impacts in a manner that can only be inferred indirectly from noise exposure contours.

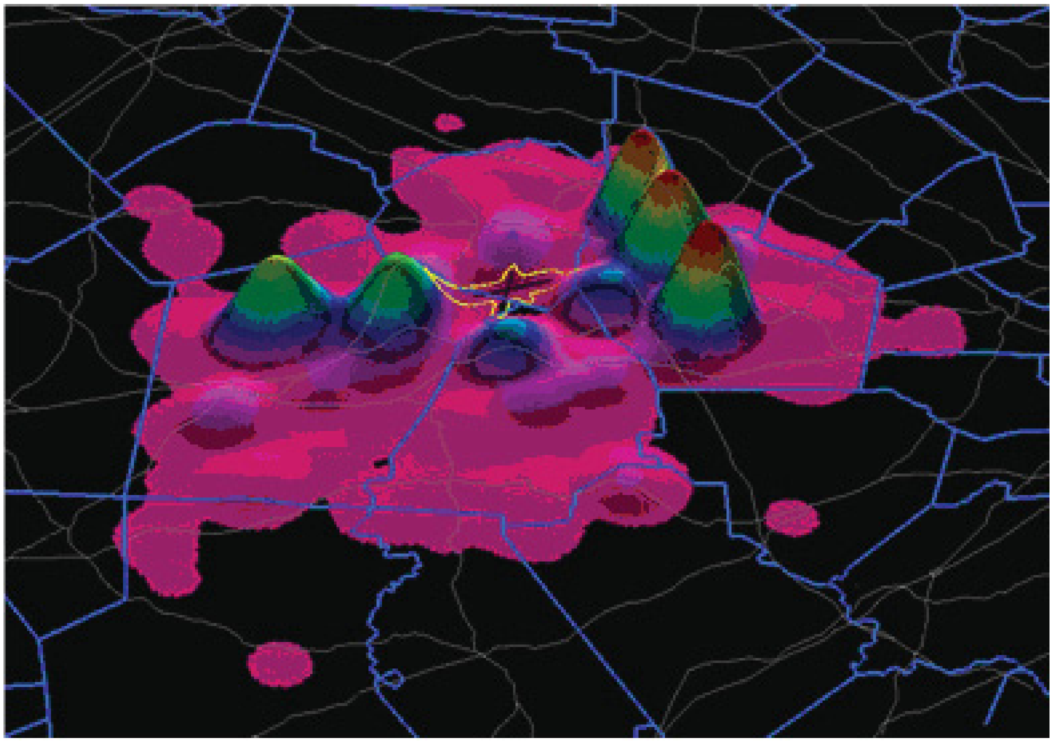


Figure 8.7. Pseudo-terrain contours for Hanscom Field, Massachusetts, showing "mountains" of complaints in towns adjacent to the field. [This photo in color is on the book's website, mentioned in the preface.]

System design

Social Security's "Future Process". Unusual among psychological projects at BBN, because of the circumstances of its origination, its size, and the manner of implementation, was one undertaken for the U.S. Social Security Administration. The idea for the project originated with a Social Security Administration employee, Richard Gonzales, who had attended Pew's University of Michigan summer course in Human Engineering. Gonzales gave BBN a small ($20k) contract to prepare a survey of attitudes among approximately 1,000 SSA claims and service representatives toward computers. Analysis of the resulting data led to a Request for Proposal from the SSA for investigation of a variety of human factors issues relating to what was envisioned to be SSA's next generation of computing systems — referred to as "The Future Process" — which was to include district office connectivity. The RFP called for the implementation of a laboratory at SSA headquarters in Baltimore, MD.

The BBN proposal for this project was written by Grignetti, Dan Massey, D. Miller and Pew. It articulated a vision of what bit-mapped, direct-manipulation interfaces were going to look like and proposed to use Xerox Altos as the experimental workstations and a DEC System 20 as the server. The contract of somewhat over $2 million was awarded to BBN and the project was launched in 1978. A laboratory was established at SSA headquarters, and staffed with seven people; Pew, who was the principal investigator relocated to Baltimore for the year-and-a-half duration of the project. Miller managed the Cambridge-based contingent of the team.

Two formal experiments were run with SSA staff from all over the country and task analyses performed on the results. Other aspects of the work included a cost-benefit analysis (Massey), development of a method for simulating interview responses of a client for use for training claims representatives (Feurzeig,), and usability testing — before there was an established field of same (Pew & Douglas Hoecker). The software was developed by the Cambridge contingent, composed of Miller, Massey, Grignetti, Lynn Bates, John Vital, and Austin Henderson. While the SSA lab was not the first usability lab ever built, Pew believes this project to be among the earliest human-centered user interface design projects that actually did iterative user testing as a part of the design process. Iterative design, in which ideas are tested with real users as they are implemented in experimental systems so the results can be used in the ongoing development process, is now widely acknowledged to be an effective and efficient approach to system development. Another aspect of this project was the training of several SSA personnel to do task analyses and to suggest interface design alternatives. Details can be found in a series of BBN reports.[115]

Military systems. Interest in human factors problems increased greatly as a consequence of the needs of the military during World War II. Topics of critical interest included the design of cockpit displays, camouflage, auditory and visual signal detection (in sonar and radar operation contexts), the effects of stress (sleep deprivation, threat, extreme environmental conditions) on human performance, vigilance, and a host of others. All military branches established research laboratories and contract research programs to meet their needs for information on such topics and the interest continued long after the war ended and to this day.

Work by BBN psychologists most directly relevant to the design of military systems or the solution of military problems included projects with the U.S. Army (BESRL) on army tactical intelligence,[116] the U.S. Navy (Naval Devices Training Center) on training for decision making,[117] and The Advanced Research Projects Agency of DoD on the human factors of command, control and communication systems.[118] All of the contracts supporting this work were won with competitive bids in response to published Requests for Proposal.

Power plant control room design and operation. Spurred by the Three-Mile Island incident, and the published opinions of experts that pointed to a variety of human factors design failures as the causal factors, we made a concerted effort to obtain work in this problem area. After a year or so with no tangible results, we received an invitation from the Electric Power Research Institute (EPRI), a research consortium funded by power companies, to become involved in a project that was investigating the introduction of computers into power plant control rooms. There was especially a need for help in understanding decision making in this context.

This was, of course, an invitation we were delighted to accept. BBN was teamed with Westinghouse. Pew was the principal investigator for BBN and his point of contact at Westinghouse was a professional colleague, David Woods, another human factors expert, with Westinghouse at the time. (Both Pew and Woods have served as president of the Human Factors and Ergonomics Society, the major professional organization for people working in this area.) Other major members of the BBN project team included Feehrer, D. Miller, and Massey.

EPRI provided the BBN team with four case studies of emergency shutdowns of nuclear plants in which critical events appeared to involve human decision making in the control room, the complete engineering analyses of the accidents, tutorial help from Westinghouse on how to interpret the analyses, and interview access to the power plant operators who were on duty when the accidents occurred. BBN's task was to analyze the decision making in the critical events and come up with general principles and guidelines for the kinds of changes in control room design, operational procedures and training that might prevent such incidents in the future. There was special interest in how computers might help.

What the BBN group produced can be considered one of the earliest cognitive task analyses and a detailed methodology for accomplishing it. Team members became experts in understanding human decision making error. Feehrer developed what came to be called "Murphy Diagrams," a type of fault-tree analysis focused on the ways that human performance could fail.[119] The final report[120] became must reading for nuclear plant engineers concerned with control room design and was cited in the *Handbook of Human Factors Engineering* as a primary data source in the field.

The Kurzweil digital piano. One of the more unusual projects undertaken by BBN psychologists involved participation in the design of the Kurzweil digital piano. In Pew's words:

> One day I received an unsolicited call from a representative of Kurzweil asking if I would be willing to undertake the product design, packaging and human factors on a new product they were designing, a digital music synthesizer having as a goal that it would have sound indistinguishable from a Steinway piano, as well as other more typical synthesizer music sounds, and cost $1,000. It was the brainchild of Raymond Kurzweil, who was himself an accomplished pianist. There were some digital synthesizers around at the very high end, but nothing really addressing the commercial market at the time.
>
> I said I would be enthusiastic about doing the human factors (the conceptual understanding of how the synthesizer would be used, the design and layout of the control panel to accommodate both set-up and performance time interactions), but would have to get a team member who could do the 'packaging' and product design (the mechanical design and materials specification of the synthesizer housing and internals and associated components that would be housed in it). They agreed and I found Paul Brefka of Latham Brefka Associates in Boston to be my partner. I don't remember the exact cost of the BBN contract but it was in the range of $60K. They had at first suggested we take payment in the form of Kurzweil Music Co. stock, but we rejected that suggestion out of hand.[s]
>
> The six month project proceeded as a true team effort and was a fine example of iterative design with participation from Kurzweil software gurus, electrical engineers (one of whom was a knowledgeable musician), and a rock musician as user representative. Carl Feehrer and I were the BBN participants. The team met weekly to review progress and hand out assignments for the next round. We delivered a final specification fully expecting to be involved in evaluating implementation and field testing, but they said "Thank you," and we never heard from them again. This is not atypical in consulting assignments. I have seen the finished product in use and as far as I can tell they implemented our specifications reasonably closely. The final cost of each synthesizer was $10,000 and the piano simulation was extremely good, but I suspect that in a blind listening test you could tell it from a Steinway.

[s]The first author confesses to arguing that BBN should take payment for this project in Kurzweil Music Company stock and being decisively overruled.

8.7 Concluding comment

BBN had, we think, an extraordinarily productive program of research in psychology, broadly defined, during the period that this chapter covers. The pace and standards were set by the first few to arrive, and in particular Licklider. They were maintained by the second group, and notably Swets. Those of us who came later were inspired, if somewhat intimidated, by the level of aspiration that had been established and were motivated to measure up as well as we could. Others can judge the extent to which we succeeded or failed in this regard. For our part, we always were more than happy to tell professional colleagues that we worked at BBN, and never found it to be disadvantageous to do so.

Acknowledgments

The authors gratefully acknowledge the assistance of Allan Collins, Carl Feehrer, Dedre Gentner, David Getty, Bill Huggins, Richard Pew, Edward Smith, John Swets, and Yvette Tenney in providing and checking facts for this account; and we thank Mary (Molly) Potter, of MIT, for graciously reading and providing comments on a draft of the manuscript.

References

1. Swets, J. A., & Markowitz, J. (1969). Factors affecting the slope of empirical ROC curves: Comparison of binary and rating response. *Perception and Psychophysics, 2*, 91-100.

2. Nickerson, R. S. Markowitz, J., & Collins, A. (1969). Effects of uncertain warning signals on reaction time. *Perception & Psychophysics, 5*, 107-112.

3. Swets, J. A., Markowitz, J., & Franzen, O. (1969). Vibrotactile signal detection. *Perception and Psychophysics, 6*, 83-88.

4. Triggs, T. J. (1971). Design criteria for space vehicle displays. BBN Report No. 2281. Submitted to Jet Propulsion Laboratories, Pasadena, CA; Levison, W. H., Triggs, T. J., & Tanner, R. (1973). Evaluation of tactual displays for flight control. BBN Report No. 2569. Submitted to Sanders Associates, Nashua, NH.

5. Pickett, R. M., & Triggs, T. J. (Eds.) (1975). *Human factors in health care.* Lexington, MA: D. C. Heath.

6. Quillian, M. R. (1969). The teachable language comprehender: A simulation program and theory of language. *Communications of the ACM, 12*, 459-476.

7. Carbonell, J. R. & Collins, A. (1973) Natural semantics in Artificial Intelligence. In *Proceedings of the Third International Joint Conference on Artificial Intelligence* (pp. 344-351). Stanford University; Collins, A., Carbonell, J. R. & Warnock, E. H. (1975). Semantic inferential processing by computer. In J. Rose (Ed.), *Advances in cybernetics and systems.* London: Gordon & Breach.

8. Huggins, A, W. F., & Nickerson, R. S. (1985). Speech quality evaluation using phoneme-specific sentences. *Journal of the Acoustical Society of America, 77*, 1896-1906; Viswanathan, V., Makhoul, J., & Huggins, A. W. F. (1978). Speech compression and evaluation. BBN Report No.3798. Submitted to DARPA.

9. Huggins, A. W. F., Karnofsky, K. F., Krasner, MA, & Wolf, J. J. (1984). ISIS (Interactive Speaker Identification System) system design. BBN Report No. 5741.

10. Fidell, S., Pearsons, K, & Bennett, R. (July, 1972). Predicting aural detectability of aircraft in noise backgrounds. U.S. Air Force Flight Dynamics Laboratory Technical Report AFFDL-TR-72-16; Pearsons, K., & Bennett, R. L. (1971). Effects of temporal and spectral combinations on the judged noisiness of aircraft sounds. *Journal of the Acoustical Society of America, 40*, 1076-1082; Pearsons, K., Bennett, R., & Fidell, S. (1971). Study of the effects of the doppler shift on perceived noisiness. Contractors Report-1779, NASA Langley; Pearsons, K., Bishop, D. E., & Horonjeff, R. D. (1969). Judged noisiness of modulated and multiple tones in broadband noise. *Journal of the Acoustical Society of America, 45*, 742-750; Pearsons, K., Fidell, S., & Bennett, R. L (1974). Prediction of aural detectability of noise signals. *Human Factors, 16*, 373-383.

11. Pearsons, K. (1977). Effect of tone/noise combination on speech intelligibility. *Journal of the Acoustical Society of America, 61*, 883-886; Pearsons, K., Williams, C. E., & Hecker, M. H. L. (1971). Speech intelligibility in the presence of time-varying aircraft noise. *Journal of the Acoustical Society of America, 50*, 426-434.

12. Fidell, S. (1978). Effectiveness of audible warning signals for emergency vehicles. *Human Factors, 20*, 19-26.

13. Fidell, S., & Teffeteller, S. (1981). Scaling the annoyance of intrusive sounds. *Journal of Sound and Vibration, 78*, 291-298; Fidell, S., Teffeteller, S., Horonjeff, R., & Green, D. M. (1979). Predicting annoyance from detectability of low level sounds. *Journal of the Acoustical Society of America, 66*, 1427-1434.

14. Fidell, S. (1980). A modern psychophysical procedure for assessing noise-induced annoyance. *Noise Control Engineering, 14*, 127-131.

15. Getty, D. J., & Howard, J. H., Jr. (Eds.) (1981). *Auditory and visual pattern recognition.* Hillsdale, NJ: Erlbaum.

16. Swets, J. A., & Pickett, R. M. (1982). *Evaluation of diagnostic systems: Methods from signal detection theory.* New York: Academic Press.

17. Green, D. M., & Swets, J. A. (1966). *Signal detection theory and psychophysics.* New York: Wiley.

18. Swets, J. A. (1984). Mathematical models of attention. In R. Parasuraman & R. Davies (Eds.), *Varieties of attention* (pp. 183-242). New York: Adademic Press; Swets, J. A., & Birdsall, T, G. (1967). Deferred decision in human signal detection: A preliminary experiment. *Perception and Psychophysics, 2*, 15-28; Swets, J. A., & Birdsall, T. G. (1978). Repeated observation of an uncertain signal. *Perception and Psychophysics, 23*, 269-274.; Swets, J. A., Green, D. M., Getty, D. J., & Swets, J. B. (1978). Signal detection and identification at successive stages of observation. *Perception and Psychophysics, 23*, 275-289; Swets, J. A., & Kristofferson, A. B. (1970). Attention. *Annual Review of Psychology, 21*, 339-366.

19. Nickerson, R. S. (1968). Response time to the second of two signals following varied versus constant intersignal intervals. *Perception & Psychophysics, 4*, 78-80; Nickerson, R. S. (1970). The effect of preceding and following auditory stimuli on response times to visual stimuli. In A. F. Sanders (Ed.), *Attention and performance III* (pp. 5-20). Amsterdam: North-Holland Publishing; Nickerson, R. S. (1971). "Same"-"different" response times: A further test of a "counter and clock" model. *Acta Psychologica, 35*, 112-117.

20. Nickerson, R. S. (1978). On the time it takes to tell things apart. In J. Requin (Ed.), *Attention and Performance VII* (pp. 77-88). Hillsdale, NJ: Erlbaum; Nickerson, R. S., & Pew, R. W. (1972). Adaptive stimulus programming estimates of the temporal indifference interval. *Perceptual and Motor Skills, 34*, 575-583; Pew, R. W. (1971). Human information processing and reaction time. BBN Report No. 2111. Submitted to the U.S. Air Force Office of Scientific Research.

21. Nickerson, R. S. (1975). Effects of correlated and uncorrelated noise on visual pattern matching. In P. M. A. Rabbitt & S. Dornic (Eds.), *Attention and performance V* (pp. 655-668). New York: Academic Press; Nickerson, R. S., & Pew, R. W. (1973). Visual pattern matching: An investigation of some effects of decision task, auditory codability and spatial correspondence. *Journal of Experimental Psychology, 98*, 36-43.

22. Fidell, S., Pearsons, K., Grignetti, M., & Green, D. M. (1970). The Noisiness of impulsive sounds. *Journal of the Acoustical Society of America 48*, 1304-1310.

23. Kalikow, D. N, Stevens, K. N., & Elliott, L. L. (1977). Development of a test of speech intelligibility in noise using sentence materials with controlled word predictability. *Journal of the Acoustical Society of America, 61*, 1337-1351.

24. Pearsons, K., Bennett, R. L., & Fidell, S. (1977). Speech levels in various noise environments U.S. Environmental Protection Agency, Environmental Health Effects Research Series, EPA-60011-77-025.

25. Pearsons, K., Fidell, S., & Teffeteller, S. (1979). *Initial study on the effects of transformer and transmission line noise on people, Volume 1: Annoyance*, EPRI EA-1240. Prepared for the Electric Power Research Institute, Palo Alto, CA.

26. Fidell, S., Silvati, L., Pearsons, K. & Green, D. M. (1998) Noticeability of a decrease in aircraft noise. *Noise Control Engineering Journal, 46*, 49-56.

27. Fidell, S., Pearsons, K., Silvati, L., & Sneddon, M (1996). An assessment of commuter aircraft noise impact. BBN Report No. 8112. Submitted NASA Langley; Pearsons, K., Howe, R., Sneddon, M., Silvati, L., & Fidell, S. (1997). *Comparison of predictors of the annoyance of commuter, Stage II and Stage III aircraft overflights as heard outdoors.* BBN Report 8201. Submitted to NASA Langley.

28. Pearsons, K. (1978). The effect of time-varying traffic noise on speech communication and annoyance. *Noise Control Engineering.* 10, 108-119.

29. Pearsons, K., Fidell, S., Silvati, L., Tabachnick, B., Howe, R., Knopf, R. C., Gramann, J., & Buchanan, T. (1992). *Short-term effects of aircraft overflights on outdoor recreationists in three wildernesses.* NPOA Report No. 91-2, National Park Service, Denver Service Center; Pearsons, K., Fidell, S., Silvati, L., Tabachnick, B., Howe, R., Knopf, R. C., Gramann, J., & Buchanan, T. (1996). Effects of aircraft overflights on wilderness recreationists. *Journal of the Acoustical Society of America, 100*, 2909-2918.

30. Fidell, S., Horonjeff, R., Schultz, T., & Teffeteller, S. (1983). Community response to blasting. *Journal of the Acoustical Society of America, 74*, 888-893.

31. Fidell,S., Horonjeff, R., & Green, D. M. (1981). Statistical analyses of urban noise. *Noise Control Engineering, 16(2)*, 75-80.

32. Fidell, S., Pearsons, K., Tabachnick, B. G., & Howe, R. (2000). Effects on sleep disturbance of changes in aircraft noise near three airports. *Journal of the Acoustical Society of America, 107*, 2535- 2547; Fidell, S., Pearsons, K., Tabachnick, B. G., Howe, R., Silvati, L., & Barber, D. S. (1995). Field Study of Noise-Induced Sleep Disturbance, *Journal of the Acoustical Society of America, 98*, 1025-1033; Pearsons, K. S., Barber, D. S., Tabachnick, B. G. and Fidell, S. (1965) Predicting noise-induced sleep disturbance. *Journal of the Acoustical Society of America*, 97. 331–338.

33. Horonjeff, R., Fidell, S., Teffeteller, S., and Green, D. M. (1982). Behavioral awakening as functions of duration and detectability of noise intrusions in the home. *Journal of Sound and Vibration, 84*, 327-336.

34. Pearsons, K., Fidell, S., Tabachnick, B., & Barber, D. S. (1995). Predicting noise-induced sleep disturbance. *Journal of the Acoustical Society of America, 97*, 331-338.

35. Fidell, S., Pearsons, K., Silvati, L., & Sneddon, M. (2002). Relationship between low-frequency aircraft noise and annoyance due to rattle and vibration. *Journal of the Acoustical Society of America, 111*, 1743-1750; Fidell, S., Silvati, L., & Pearsons K. (2002) Relative rates of growth of annoyance of impulsive and non-impulsive noises. *Juornal of the Acoustical Society of America, 111*, 576-585.

36. Green, D .M. Kidd, G., Jr., & Stevens, K. N. (1987). High-frequency audiometric assessment of a young adult population. *Journal of the Acoustical Society of America, 81*, 485-494; Stevens, K. N., Berkovitz, R., Kidd, G. Jr., & Green, D. M. (1987). Calibration of ear canals for audiometry at high frequencies. *Journal of the Acoustical Society of America, 81*, 470-484.

37. Collins, A. & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of Verbal*

Learning and Verbal Behavior, 8, 240-247; Collins, A. & Quillian, M. R. (1970). Does category size affect categorization time? *Journal of Verbal Learning and Verbal Behavior, 9*, 432-438; Collins, A. & Quillian, M. R. (1970b). Facilitating retrieval from semantic memory: The effect of repeating part of an inference. *Acta Psychologica, 33*, 304-314; Collins, A. & Quillian, M. R. (1972). Experiments on semantic memory and language comprehension. In L. W. Gregg (Ed.), *Cognition in learning and memory* (pp. 117-138). New York: Wiley; Collins, A. & Quillian, M. R. (1972). How to make a language user. In E. Tulving and W. Donaldson (Eds.), *Organization of memory* (pp. 309-351). New York: Academic Press.

38. Nickerson, R. S. (1972). Auditory codability and the short-term retention of visual information. *Journal of Experimental Psychology, 95*, 429-436; Nickerson, R. S. (1976). Short-term retention of visually presented stimuli: Some evidence of visual encoding. *Acta Psychologica, 40*, 153-162; Nickerson, R. S. (1976). Visual memory for letters and random dot patterns: Could eye movement data help us understand it better? In R. A. Monty & J. W. Senders (Eds.), *Eye movements and psychological processes* (pp. 515-521). New York: Wiley.

39. Gentner, D. (1981). Integrating verb meanings into context. *Discourse Processes*, 4, 349-375; Gentner, D. (1981). Verb semantic structures in memory for sentences: Evidence for componential representation. *Cognitive Psychology, 13*, 56-83.

40. Gentner, D. (1982). Why nouns are learned before verbs: Linguistic relativity versus natural partitioning. In S. A. Kuczaj (Ed.), *Language development: Vol. 2. Language, thought and culture* (pp. 301-334). Hillsdale, NJ: Erlbaum; Nickerson, R. S. (1977). Crossword puzzles and lexical memory. In S. Dornic (Ed.), *Attention and performance VI* (pp. 699-718). Hillsdale, NJ: Erlbaum..

41. Nickerson, R. S., & Adams, M. J. (1979). Long-term memory for a common object. *Cognitive Psychology, 11*, 287-307.

42. Nickerson, R. S. (1980). Motivated retrieval from archival memory. In H. E. Howe, Jr. & J. H. Flowers (Eds.), *Motivation and cognitive processes: 1980 Nebraska Symposium on Motivation* (pp. 73-119). Lincoln, NE: University of Nebraska Press.

43. Nickerson, R. S. (1984). Retrieval inhibition from part-list cuing: A persisting enigma in memory research. *Memory and Cognition, 12*, 531-552; Nickerson, R. S., Smith, E. E., & Wallach, R. W. (1981). *Memory search of semantic categories following exposure to category instances.* BBN Report No. 4822. Submitted to the National Institute of Education.

44. Collins, A. (1985). Component models of physical systems. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society* (pp. 80-89). Hillsdale, NJ: Erlbaum; Collins, A. & Gentner, D. (1982). Constructing runnable mental models. In *Proceedings of the Fourth Annual Cognitive Science Society* (pp. 86-89). Hillsdale, NJ: Erlbaum; Collins, A. & Gentner, D. (1983). Multiple models of evaporation processes. In *Proceedings of the Fifth Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum; Collins, A. & Gentner, D. (1987). How people construct mental models. In D. Holland and N. Quinn (Eds.), *Cultural models in thought and language* (pp. 243-265). Cambridge UK: Cambridge University Press; Ablex; Collins, A. & Smith, E. E. (1984). Applied cognitive science. In H. W. Stevenson & Q. Jing (Eds.), *Issues in Cognition* (pp. 447-465). Washington, DC: National Academy of Sciences; Gentner, D., & Stevens, A. L. (Eds.) (1983). *Mental models.* Hillsdale, NJ: Erlbaum; Smith, E. E., & Collins, A. (1981). Use of goal-plan knowledge in understanding stories. *Proceedings of the Third Annual Conference of the Cognitive Science Society*, Berkeley, CA; Stevens, A. & Collins, A. (1980). Multiple conceptual models of a complex system. In R. Snow, P. Federico, & W. Montague (Eds.), *Aptitude learning and instruction: Cognitive processing analysis* (pp. 177–197). Hillsdale, NJ: Erlbaum; Tenney, Y., & Gentner, D. (1984). What makes analogies accessible: Experiments in the water-flow analogy for electricity. *Proceedings of the International Workshop on Research Concerning Students' Knowledge of Electricity*, Ludwigsburg, Germany.

45. Frederiksen, J. R., & White, B. (1989). Mental models and understanding. A problem for science education. In E. Scanlon & T. O'Shea (Eds.), *New diresctions in educational technology*

(pp. 211-226). New York: Springer Verlag; White, B. & Frederiksen, J. R. (1987). Qualitative models and intelligent learning environments. In R. Lawler & M. Yazadani (Eds.), *AI and education* (pp. 281-305). Norwood, NJ: Ablex; White, B., & Frederiksen, J. R. (1989). Causal models as intelligent learning environments for science and engineering education. *Applied Artificial Intelligence, 3*, 167-190.

46. Baker, M., Burstein, M. H., & Collins, A. (1987). Implementing a model of human plausible reasoning. In *Proceedings of the International Joint Conference on Artificial Intelligence* (pp. 185-188). San Mateo, CA: Morgan Kaufmann; Collins, A. (1978). Fragments of a theory of human plausible reasoning. In D. Waltz (Ed.), *Proceedings of Conference on Theoretical Issues in Natural Language Processing 2.* (pp. 194-201). Champaign-Urbana: University of Illinois; Collins, A., Burstein, M. H., & Baker, M. (1991). Plausible generalization: Extending a model of human plausible reasoning. *Journal of the Learning Sciences, 1*, 319-359; Collins, A., Carbonell, J. R. & Warnock, E. H. (1975). Semantic inferential processing by computer. In J. Rose (Ed.), *Advances in cybernetics and systems.* London: Gordon & Breach; Collins, A., & Michalski, R. S. (1989). The logic of plausible reasoning: A core theory. *Cognitive Science*, 13, 1-49; Collins, A., Warnock, E. H., Aiello, N., & Miller, M. L. (1975). Reasoning from incomplete knowledge. In D. Bobrow & A. Collins (Eds.), *Representation and understanding* (pp. 383-415). New York: Academic Press; Gentner, D., & Collins, A. (1981). Inference from lack of knowledge. *Memory and Cognition, 9*, 434-443; Rips, L. J. & Collins, A. (1993). Categories and resemblance. *Journal of Experimental Psychology: General, 122*, 468-486.

47. Gentner, D. (1980). *The structure of analogical models in science.* BBN Report No. 4451. Submitted to the Office of Naval Research; Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science, 7*, 155-170.

48. Forbus, K. D., & Gentner, D. (1986). Causal reasoning about quantities. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society* (pp. 196-207); Forbus, K. D., & Gentner, D. (1986). Learning physical domains: Toward a theoretical framework. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach* (Vol. 2, pp. 311-348). Los Altos, CA: Kaufmann; Gentner, D. (1988). Metaphor as structure mapping: The relational shift. *Child Development, 59*, 47-59; Gentner, D., & Gentner, D. R. (1983). Flowing waters or teeming crowds: Mental models of electricity. In D. Gentner & A. L. Stevens (Eds.), *Mental models* (pp. 99-129). Hillsdale, NJ: Erlbaum; Gentner, D., & Toupin, C. (1986). Systematicity and surface similarity in the development of analogy. *Cognitive Science, 10*, 277-300.

49. Adams, M. J. (1980). Inductive deductions and deductive inductions. In R. S. Nickerson (Ed.), *Attention and performance VIII* (pp. 745-762). Hillsdale, NJ: Erlbaum; Nickerson, R. S. (1986). Reasoning. In R. F. Dillon & R. J. Sternberg, (Eds.), *Cognition and instruction* (pp. 343-373). New York: Academic Press; Nickerson, R. S. (1986). *Reflections on Reasoning.* Hillsdale, NJ: Erlbaum. Rosebery, A. S., & Rubin, A. D. (1989). Reasoning under uncertainty: Developing statistical reasoning. *Journal of Mathematical Behavior, 8*, 205-221; Rosebery, A. S., & Rubin, A. D. (1990). Teaching statistical reasoning with computers. *Teaching Statistics, 12*(2), 38-42.

50. Adams, M. J. (1979). *Differences between good and poor readers.* BBN Report No. 3993. Submitted to the National Institute of Education; Adams, M. J. (1979). Models of word recognition. *Cognitive Psychology, 11* 133–176; Adams, M. J. (1980). Failure to comprehend and levels of processing in reading. In R. Spiro, B. Bruce and W. Brewer, *Theoretical issues in reading comprehension.* Hillsdale, NJ: Erlbaum; Adams, M. J. (1981). What good is orthographic redundancy? In O. J. L. Tzeng & H. Singer (Eds.), *Perception of print: Reading research in experimental psychology* (pp. 197-221). Hillsdale, NJ: Erlbaum; Adams, M. J., & Collins, A. (1979). *A schema-theoretic view of reading.* BBN Report No. 3548. Submitted to the National Institute of Education; Bruce, B. C. (1981). *A social interaction model of reading.* (ERIC Document Reproduction Service No. ED 208 363). Collins, A., & Haviland, S. E. (1979). Children's reading problems. In R. W. Tyler (Ed.), *Testing, teaching and learning* (pp. 136-145). Washington, DC: National Institute of Education; Collins, A., & Smith, E. E. (1982). Teaching the process of reading comprehension. In D. K. Detterman & R. J. Sternberg (Eds.), *How and how much can intelligence be increased?* (pp. 173-186). Norwood, NJ: Ablex; Frederiksen, J. R., & Warren, B. M. (1987). A cognitive framework for developing expertise in reading. In R. Glaser (Ed.), *Advances in instructional psychology*, Vol 111 (pp. 1-38). Hillsdale, NJ: Erlbaum; Frederiksen, J. R., Weaver, P. A., Warren, B. M., Gillotte, H. P., & Rosebery, A. S. (1984). A componential approach to training reading skills. (ERIC Document Reproduction Service No. ED 229 727); Gentner, D. (1981). Integrating verb meanings into context. *Discourse Processes, 4*, 349-375; Gentner, D. (1981). Some interesting differences between nouns and verbs. *Cognition and Brain Theory*, 4, 161-178; Huggins, A. W. F. (1977). *Syntactic aspects of reading comprehension.* (ERIC Document Reproduction Service No. ED 142 972); Nickerson, R. S. (1981). Speech understanding and reading: Some differences and similarities. In O. J. L. Tzeng & H. Singer (Eds.), *Perception of print: Reading research in experimental psychology* (pp. 257-289). Hillsdale, NJ: Erlbaum; Rubin, A. D., Bruce, B. C., & Brown, J. S. (1976). *A process-oriented language for describing aspects of reading comprehension.* (ERIC Document Reproduction Service No. ED 136 188).

51. Bruce, B. C., Collins, A., Rubin, A. D., & Gentner, D. (1978). *Cognitive science approach to writing.* (ERIC Document Reproduction Service No. ED 157 039); Collins, A., & Gentner, D. (1980). A framework for a cognitive theory of writing. In L. W. Gregg & E. Steinberg (Eds.), *Cognitive processes in writing: An interdisciplinary approach* (pp. 51-72). Hillsdale, NJ: Erlbaum; Rubin, A. D., & Hansen, J. (1983). Reading and writing: How are the first two "R's" related? In J. Orasanu (Ed.), *A decade of reading research: Implications for practice* (pp. 163- 170). Hillsdale, NJ: Erlbaum.

52. Bruce, B. C., Rubin, A. D., & Collins, A., & Gentner, D. (1982). Three perspectives on writing. *Educational Psychologist, 17*, 131-145; Bruce, B. C., Michaels, S., & Watson-Gegeo, K. (1985). How computers can change the writing process. *Language Arts, 62*, 143-149; Rubin, A. D. (1983). The computer confronts language arts: Cans and shoulds for education. In A. C. Wilkinson (Ed.), *Classroom computers and cognitive science* (pp. 201-271). New York: Academic Press; Rubin, A. D., & Bruce, B. C. (1985). QUILL: Reading and writing with a microcomputer. In B. A. Hutson (Ed.), *Advances in reading and language research* (pp. 97-117). Greenwich, CT: JAI Press; Rubin, A. D., & Bruce, B. C. (1986). Learning with QUILL: Lessons for students, teachers and software designers. In T. E. Raphael (Ed.), *Contexts of school based literacy* (pp. 217-230).

53. Adams, M. J., & Huggins, A. W. F. (1985). The growth of children's sight vocabulary: A quick test with educational and theoretical implications. *Reading Research Quarterly, 20*, 262-281; Adams, M. J., Huggins, A. W. F., Starr, B. J., Rollins, A. M., Zuckerman, L. E., Stevens, K. N., & Nickerson, R. S. (1980). *A prototype test of decoding skills.* BBN Report No. 4316. Submitted to the National Institute of Child Health and Human Development.

54. Smith, E. E., & Goodman, L. (1982). *Understanding instructions: The role of explanatory material*l. BBN Report No. 5088. Submitted to the Office of Naval Research; Smith, E. E., & Spoehr, K. (1987). *Basic processes and individual differences in understanding and using instructions.* BBN Report No. 6029. Submitted to the Office of Naval Research.

55. Collins, A. (1977). Processes in acquiring knowledge. In R. C. Anderson, R. J. Spiro, & W. E. Montague (Eds.), *Schooling and the acquisition of knowledge* (pp. 339-363). Hillsdale, NJ: Erlbaum; Collins, A. (1985). Teaching reasoning skills. In S. Chipman, J. Segal, & R. Glaser (Eds.), *Thinking and learning skills: Current research and open questions* (Vol. 2, pp. 579-586). Hillsdale, NJ: Erlbaum; Collins, A. (1987). A sample dialogue based on a theory of inquiry teaching. In C. M. Reigeluth (Ed.) *Instructional theories in action: Lessons illustrating selected theories and models* (pp. 181-199). Hillsdale, NJ: Erlbaum; Collins, A. (1988). Different goals of inquiry teachers. *Questioning Exchange, 2*, 39-45; Collins, A., & Stevens, A. L. (1982). Goals and strategies of inquiry teachers. In R. Glaser (Ed.). *Advances in instructional psychology* (Vol. 2, pp. 65-119). Hillsdale, NJ: Erlbaum; Collins, A., & Stevens, A. L. (1983). A cognitive theory of interactive teaching. In C. M. Reigeluth (ed.), *Instructional design theories and models: An overview* (pp. 247-278). Hillsdale, NJ: Erlbaum.

56. Collins, A. (1991). Cognitive apprenticeship and instructional technology. In L. Idol & B. F. Jones (Eds.) *Educational values and cognitive instruction: Implications for reform* (pp. 119-136). Hillsdale, NJ: Erlbaum; Collins, A., Brown, J. S., & Holum, A. (1991). Cognitive apprenticeship: Making thinking visible. *American Educator, 15*, 6-11, 38-46; Collins, A., Brown, J. S., & Newman,

S. E. (1988). A framework for cognitive apprenticeship. *Impact on Instructional Improvement, 22*, 33-39; Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 453-494). Hillsdale, NJ: Erlbaum; Collins, A., Hawkins, J., & Carver, S. M. (1991). A cognitive apprenticeship for disadvantaged students. In B. Means, C. Chelemer & M..S. Knapp (Eds.), *Teaching advanced skills to at-risk students* (pp. 216-243). San Francisco: Jossey-Bass.

57. Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher, 18*, 32-42.

58. Adams, M. J. (1981). *What is intelligence?* BBN Report No. 4801. Submitted to the National Institute of Education; Feurzeig, W., Horwitz, P., & Nickerson, R. S. (1981). *Microcomputers in education.* BBN Report No. 4798. Submitted to the National Institute of Education; Nickerson, R. S. (1982). *Three uses of computers in education.* BBN Report No. 5178. Submitted to the National Institute of Education; Nickerson, R. S., Perkins, D. N., & Smith, E. E. (1985). *The teaching of thinking.* Hillsdale, NJ: Erlbaum; Nickerson, R. S., Salter, W., Shepard, S., & Herrnstein, J. (1984). *The teaching of learning strategies.* BBN Report No. 5578. Submitted to the National Institute of Education; Smith, E. E., & Bruce, B. C. (1981). *An outline of a conceptual framework for the teaching of thinking skills.* BBN Report No. 4844. Submitted to the National Institute of Education.

59. Nickerson, R. S., & Zodhiates, P. P. (Eds.). (1988). *Technology in education: Looking toward 2020.* Hillsdale, NJ: Erlbaum.

60. Harvard University and Bolt Beranek and Newman Inc. (1983). *Project Intelligence: The development of procedures to enhance thinking skills* (Final Report). Submitted to The Minister for the Development of Human Intelligence, Republic of Venezuela.

61. Adams, M. J. (1989). Thinking skills curricula: Their promise and progress. *Educational Psychologist, 24*, 25-77; Chance, P. (1986). *Thinking in the classroom.* New York: Teachers College Press; Herrnstein, R. J., Nickerson, R. S., Sanchez, M, & Swets, J. A. (1986). Teaching thinking skills. *American Psychologist, 41*, 1279-1289; Nickerson, R. S. (1994). Project Intelligence. In R. J. Sternberg, S. J. Ceci, J. Horn, E. Hunt, J. D. Matarazzo, & S. Scarr (Eds.), *Encyclopedia of intelligence* (pp. 857-860). New York: MacMillan; Perkins, D. N. (1995). *Outsmarting IQ: The emerging science of learnable intelligence.* New York: The Free Press.

62. Adams, M. J. (Ed). (1986). *Odyssey: A curriculum for thinking.* Watertown, MA: Mastery Education Corporation.

63. Nickerson, R. S., Baron, S, Collins, A., & Crothers, C. G. (1968). *Investigation of some of the problems of vehicle rear lighting.* BBN Report No. 1586. Submitted to National Highway Safety Bureau.

64. Galloway, W. J. (1971). *Motor vehicle noise: Identification and analysis of situations contributing to annoyance.* BBN Report 2082. Submitted to the Motor Vehicle Manufacturers Association; Jones, G. (1971). *A study of annoyance from motor vehicle noise.* BBN Report 2112. Submitted to the Motor Vehicle Manufacturers Association.

65. Fidell, S., Jones, G., & Pearsons, K. S. (1973). *Feasibility of a novel technique for assessing noise-induced annoyance.* BBN Report No. 2423. Submitted to the U..S. Department of Transportation.

66. Swets, J., A., Franken, P. A., Fidell, S., & Jones, G. (1970). *Human response to sonic booms: A research program plan.* BBN Report No. 1831. Submitted to the U.S. Department of Transportation-Federal Aviation Agency.

67. Fidell, S., & Jones, G. (1975) Effects of cessation of late-night flights on an airport community. *Journal of Sound and Vibration, 42*, 411-427.

68. Fidell, S., Silvati, L, & Pearsons, K. (1996) *Social survey of community preferences for aircraft noise mitigation measures.* BBN Report 8172.

69. Schultz, T. J. (1978). Synthesis of social surveys on noise annoyance. *Journal of the Acoustical Society of America, 64*, 377-405.

70. Jones, G. (1969). A survey of usage of children's car seats and related devices. BBN Report No. 1829. Submitted to the Juvenile Products Manufacturers Association; Jones, G. (1970). *A survey of the reasons that children use car seats.* BBN Report No. 1976. Submitted to the Juvenile Products Manufacturers Association.

71. Jones, G. (1973). *A survey of mothers' experiences with full-sized cribs and related furniture that small children sleep in.* BBN Report No. 2521. Submitted to the Juvenile Products Manufacturers Association.

72. Miller, D. C., Grignetti., & Jones, G. (1972). *A Comparison of time-sharing and batch processing methods for COBOL software development at four Navy programming facilities.* BBN Report No. 2307. Submited to the U.S. Navy, Office of Naval Research; Pickett, R., Brown, D. W., Jones, G., & Massey, L. D. (1974). *Information commerce in an ambulatory clinic: Study plan.* BBN Report 2903. Submitted to the U.S. Department of Health Education and Welfare — Health Resources.

73. Tenney, Y. J., Rogers, W. H., & Pew, R. W. (1998). Pilot opinions on cockpit automation issues. *The International Journal of Aviation Psychology 8*, 103-120.

74. Swets, J. A., Pickett, R. M., Whitehead, S. F., Getty, D. J., Schnur, J. A., Swets, J. B., & Freeman, B. A. (1979). Assessment of diagnostic technologies. *Science, 205*, 753-759.

75. Swets, J. A., Getty, D. J., Pickett, R. M., D'Orsi, C. J., Seltzer, S. E., & McNeil, B. J. (1991). Enhancing and evaluating diagnostic accuracy. *Medical Decision Making, 11*, 9-18.

76. Getty D. J., Pickett, R. M., D'Orsi, C. J., & Swets J. A. (1988). Enhanced interpretation of diagnostic images. *Investigative Radiology, 23*, 240-252.

77. D'Orsi, C. J., Getty, D. J., Swets, J.A , Pickett, R. M., Seltzer, S. E., McNeil, B. J. (1992). Reading and decision aids for improved accuracy and standardization of mammographic diagnosis. *Radiology, 184*, 619-622; Swets, J. A., Getty, D. J., Pickett, R. M., D'Orsi, C. J., Seltzer, S. E., & McNeil, B. J. (1991). Enhancing and evaluating diagnostic accuracy. *Medical Decision Making, 11*, 9-18.

78. Seltzer, S. E., McNeil, B. J., D'Orsi, C. J., Getty, D. J., Pickett, R. M., & Swets, J. A. (1992). Combining evidence from multiple imaging modalities: A feature-analysis method. *Computerized Medical Imaging and Graphics, 16*(6): 373-380.

79. Getty, D. J., Pickett, R. M., Chylack, L.T, Jr.., McCarthy, D. F., & Huggins, A. W. F. (1988). An enriched set of features of nuclear cataract identified by multidimensional scaling. *Current Eye Research, 8*, 1-8.

80. Seltzer, S. E., Getty, D. J., Pickett, R. M., Swets, J. .A., Sica, G., Brown, J., Saini, S., Mattrey, R. F., Harmon, B., Francis, I. R., Chezmar, J., Schnall, M. O., Siegelman, E. S., Ballerini, R., & Bhat, S. (2002). Multimodality diagnosis of liver tumors: feature analysis using CT, liver-specific contrast enhanced MR and a computer model. *Academic Radiology, 9*, 256-269.

81. Getty, D. J., Seltzer, S. E., Tempany, C. M. C., Pickett, R. M„ Swets, J. A., & McNeil, B. J. (1997). Prostate cancer: Relative effects of demographic, clinical, histologic, and MR imaging variables on the accuracy of staging. *Radiology, 204*, 471-479; Seltzer, S. E., Getty, D. J., Tempany, C. M. C., Pickett, R. M., Schnall, M. D., McNeil, B. J., & Swets, J. A. (1997). Staging prostate cancer with MR imaging: a combined radiologist-computer system. *Radiology, 202*, 219-226.

82. Getty, D. J., Swets, J. A., & Swets, J. B. (1980). The observer's use of perceptual dimensions in signal identification. In R. S. Nickerson (Ed.) *Attention and Performance VIII* (pp. 361-380). New York: Academic Press; Getty, D. J., Swets, J. A., & Swets. J. B. (1981). Perceptual spaces revealed by multidimensional scaling procedures. In D. J. Getty & J. H. Howard, Jr.. (Eds.), *Auditory and visual pattern recognition* (pp. 168-180). Hillsdale, NJ: Erlbaum; Getty, D. J., Swets, J. A., Swets, J. B., & Green, D. M. (1979). On the prediction of confusion matrices from similarity judgments. *Perception and Psychophysics, 26*, 1-19

83. Getty, D. J. (1996). Assisting the radiologist to greater accuracy. In R.L. Van Metter & J. Beutel (Eds.), *SPIE medical imaging 1996: Physics of medical imaging* (Vol. 2708; pp. 2-15), Bellingham, WA: SPIE.

84. Getty, D. J., & Huggins A. W. F. (1986). Volumetric 3-D displays and spatial perception. In E. J. Wegman & D. J. DePriest (Eds.), *Statistical image processing and graphics* (pp 312-343). New York: Marcel-Dekker; Huggins, A. W. F., & Getty, D. J. (1982). Display-control compatibility in 3-D displays. In D. J. Getty (Ed.), *3-D displays: perceptual research and applications to military systems.* Report of the National Academy of Sciences (Committee on Human Factors), available as Ms. 2570, Select Press, San Rafael, CA.

85. D. J. Getty, (Ed.), *3-D displays: Perceptual research and applications to military systems.* Report of the National Academy of Sciences, Committee on Human Factors, available as Ms. 2570, Select Press, San Rafael, CA.

86. Getty, D. J. (2002). Stereoscopic digital mammography: perceptual and display factors leading to improved early detection of breast cancer. In H-O Peitgen (Ed.), *Digital Mammography: IWDM 2002, 6th International Workshop on Digital Mammography* (pp. 431-435). Berlin: Springer; Getty, D. J., Pickett, R. M., & D'Orsi, C. J. (2001). Stereoscopic digital mammography: improving detection and diagnosis of breast cancer. In: H. U. Lemke, M. W. Vannier, K. Inamura, A. G. Farman (Eds.), & K. Doi. *Computer assisted radiology and surgery 2001; Proceedings of the 15th International Congress and Exhibition.* (pp. 506-511). Amsterdam: Elsevier.

87. Kalikow, D. N., & Swets, J. A. (1972). Experiments with computer-controlled displays in second-language learning. *IEEE transactions in Audio and Electroacoustics, AU-20*, 23-28.

88. Nickerson, R. S., Kalikow, D. N., & Stevens, K. N. (1976). Computer-aided speech training for the deaf. *Journal of Speech and Hearing Disorders, 41*, 120-132; Nickerson, R. S., & Stevens, K. N. (1973). Teaching speech to the deaf: Can a computer help? *IEEE Transactions on Audio and Electroacoustics, AU-21*, 445-455.

89. Nickerson, R. S. (1975). Characteristics of the speech of deaf persons. *The Volta Review, 77*, 342-362; Nickerson, R. S., Stevens, K. N., Boothroyd, A., & Rollins, A. (1974). *Some observations on timing in the speech of deaf and hearing speakers.* BBN Report No. 2905. Submitted to the U.S. Department of Health, Education and Welfare, Bureau of Education for the Handicapped. Stevens, K. N., Nickerson, R. S., Boothroyd, A., & Rollins, A. (1976). Assessment of nasality in the speech of deaf children. *Journal of Speech and Hearing Research, 19*, 394-416; Stevens, K. N., Nickerson, R. S., & Rollins, A. M. (1978). *On describing the suprasegmental properties of the speech of deaf children.* BBN Report No. 3955. Submitted to the U. S. Department of Health, Education and Welfare, Bureau of Education for the Handicapped.

90. Grignetti, M. Myer, T, Nickerson, R. S., & Rubenstein, R. (1977). *Computer-aided communications for the deaf.* BBN Report No. 3738.

91. Schwartz, R., Chow, Y. I. L., Kimball, O., Krasner, M., & Makhoul, J. (1984). *Research in continuous speech recognition.* BBN Report No. 5799. Submitted to the Office of Naval Research.

92. Huggins, A. W. F., Houde, R. A., & Colwell, E. (1986). *Vidvox human factors investigation.* BBN Report No. 6187. Submitted to Sensory Aids Foundation.

93. Rubinstein, R., & Rollins, A. (1978). Demonstration of the use of computer-assisted instruction with handicapped children. BBN Report No. 4049.

94. Licklider, J. C. R. (1960). Man-computer symbiosis. *Institute of Radio Engineers Transactions on Human Factors in Electronics, HFE-1*, 4-11.

95. Carbonell, J. R., Elkind, J. I., & Nickerson, R. S. (1968). On the psychological importance of time in a time-sharing system. *Human Factors, 10*, 135-142; Nickerson, R. S. (1969). Man-computer interaction: A challenge for human factors research. *Ergonomics, 12*, 501-517; Nickerson, R. S. (1981). Why interactive computer systems are sometimes not used by people who might benefit from them. *International Journal of Man-Machine Studies, 15*, 469-483; Nickerson, R. S. (1986). *Using computers: Human factors in information systems.* Cambridge, MA: MIT

Press; Nickerson, R. S., Elkind, J. I., & Carbonell, J. R. (1968). Human factors and the design of time-sharing computer systems. *Human Factors, 10*, 127-134; Nickerson, R. S., Myer, T. H., Miller, D. C., & Pew, R. W. (1981). *User-computer interaction: Some problems for human factors research.* BBN Report 4719. Submitted to the U.S. Army Research Institute; Nickerson, R. S., & Pew, R. P. (1990). Toward more compatible human-computer interfaces. *Spectrum, 27*, 40-43; Pew, R., Rollins, A., Feehrer, C., & Pickett, R. (1975). *Human factors considerations in the design of the ASCS interactive computer system.* BBN Report No. 3128. Submitted to the U.S. Department of Agriculture; Pew, R. W., Woods, W. A., Stevens, A. L., & Weene, P. (1978). *Identifying information systems requirements for decision making.* BBN Report No. 3801. Submitted to U.S. Air Force, Hanscom Air Force Base.

96. Pew, R. W., & Rollins, A. (1975a). *Dialogue specification procedures.* BBN Report No. 3092. Submitted to U.S. Department of Agriculture; Pew, R. W., & Rollins, A. (1975b). *Revised EDN-dialogue specification procedures.* BBN Report No. 3129. Submitted to U.S. Department of Agriculture.

97. Feehrer, C. E., Pew, R. W., Miller, D., & Weene, P. (1978). *Report of studies performed in support of Lincoln Lab program in secure voice conferencing.* BBN Report No. 3681. Submitted to MIT/Lincoln Laboratory; Forgie, J., Feehrer, C., & Weene, P. (1979). *Voice conferencing final report.* Tech. Rep. DDC AD-A074498. Lexington MA: MIT Lincoln Laboratory.

98. Adams, M. J., Corker, K., Feehrer, C., Fox, J., & Grignetti, M. (1985). Cockpit automation technology. BBN Report No. 5981. Submitted to NASA; Pew, R. W., Olstad, M., Sherman, H., & Adams, M. J. (1986). *Cockpit automation technology, Final report, Vol. 1.* BBN Report No. 6133. Submitted to NASA.

99. Tenney, Y. J., Pew, R. W., Rogers, W. H., & Salter, W. J. (1997). *Evaluation of Faultfinder, a decision aid for fault diagnosis in the glass cockpit.* BBN Report No. 8191. Submitted to NASA Langley.

100. Baron, S., & Feehrer, C. (1985). *An analysis of application of AI to the development of intelligent aids for flight crew tasks.* BBN Report No. 5937. Submitted to NASA.

101. MacMillan, J., Tenney, Y., & Pew, R. W. (1997). *Human error in advanced maintenance control centers.* BBN Report 8205. Submitted to Systems Resources Corporation.

102. Deutsch, S., Pew, R. W., Granville, R., & Roberts, B. (1998). *Automating maintenance instructions study, Volume 1.* BBN Report No. 8231.

103. Swets, J. A. (1995). *A primer on signal detection theory and aviation.* BBN Report 8046.

104. Getty, D. J., Swets, J. A., Pickett, R. M., & Gonthier, D. (1995). System operator response to warnings of danger: A laboratory investigation of the effects of the predictive value of a warning on human response time. *Journal of Experimental Psychology: Applied, 1*, 19-33.

105. Swets, J. A., & Getty, D. J. (1991). *Developing methods for the establishment of sensitivity and threshold requirements for human-centered decision aids.* BBN Report No. 3271. Submitted to NASA.

106. Adams, M. J., Tenney, Y. J., & Pew, R. (1991). State of-the-art-report: *Strategic workload and the cognitive management of advanced multi-task systems.* BBN Report No. 7650. Submitted to CSDFIAC.

107. Adams, M. J., Tenney, Y. J., & Pew, R. W. (1995). Situation awareness and the cognitive management of complex systems. *Human Factors, 37*, 85-104; Deutsch, S. E., Pew, R. W., Rogers, W. H., & Tenney, Y. J. (1994). Toward a methodology for defining situation awareness requirements: A progress report. BBN Report No. 7963; Tenney, Y. J., Adams, M. J., Pew, R. W., Huggins, A. W. F., & Rogers, W. H. (1992). A principled approach to the measurement of situation awareness in commercial aviation. NASA Contract Report No. 4451. Hampton, VA: NASA Langley.

108. Elkind, J. I. (1953). *Tracking response characteristics of the human operator.* Memo. No. 40, Human Factors Operations Research Laboratories, Air Research and Development Command, Washington, DC.

109. Licklider, J. C. R. (1960). Quasi-linear operator models in the study of manual tracking. In D. R. Luce (Ed.), *Developments in mathematical psychology* (pp. 167-279). Glencoe, IL: The Free Press.

110. Baron, S., Feehrer, C., Murali, R., Pew, R. W., & Horwitz, P. (1982). *An approach to modeling supervisory control of a nuclear power plant.* BBN Report No. 4811. Submitted to Oak Ridge National Laboratory.

111. Grignetti, M., Miller, L., Nickerson, R. S., & Pew, R. W. (1972). *Information processing models and computer aids for human performance, Final report, section 2, Task 2: Human-computer interaction models.* BBN Report No. 2352. Submitted to DARPA.

112. Hudlicka, E., Adams, M. J., & Feehrer, C. (1992). *Research, development, training, and evaluation support (RDTE) — Computational cognitive models: Informal technical summary phase 1.* BBN Report No. 7879. Submitted to Armstrong Laboratories, Wright-Patterson Air Force Base; Miller, D., Feehrer, C., Muralidharan, R., Pew, R. W., & Baron, S. (1978). *Development of human performance models for man-machine system simulation.* BBN Report No. 3739. Submitted to U.S. Air Force Office of Scientific Research; Pew, R. W., Feehrer, C., Baron, S., & Miller, D. (1977). *Critical review and analysis of performance models applicable to man-machine systems.* BBN Report No. 3446. Submitted to U.S. Air Force Office of Scientific Research.

113. Baron, S., Feehrer, C., & Zacharias, G. (1979). *Analysis and modeling of flight crew procedures in approach and landing.* BBN Report No. 4112. Submitted to NASA.

114. Deutsch, S. E., & Pew, R. W. (2001). Modeling human error in D-OMAR. BBN Report No. 8328. Submitted to NASA; Deutsch, S. E., & Pew, R. W. (2003). *Modeling the NASA baseline and SVS-equipped approach and landing scenarios in D-OMAR.* BBN Report No. 8364. Submitted to NASA; Feehrer, C., Deutsch, S., & Cramer, N. (1993). *Research, development, training and evaluation (RTDE) support: Operator model architecture (OMAR).* BBN Report No. 8019. Submitted to NASA.

115. Hoecker, D., & Pew, R. W. (1980). *User input to the design and evaluation of computer-assisted service delivery.* BBN Report No. 4358. Submitted to the U.S. Social Security Administration; Massey, L. D., Miller, D. C., Larkin, K. M., & Pew, R. W. (1980). *A plan for developing computer assistance in SSA district offices.* BBN Report 4356. Submitted to the U.S. Social Security Administration; Pew, R. W. (1979). *Human factors evaluation criteria for the SSA.* BBN Report No. 4042. Submitted to the U.S. Social Security Administration; Pew, R., Hoecker, D. G., Miller, D. C., & Walker, J. H. (1979). *User interface requirements for computer assisted service delivery.* BBN Report No. 4045. Submitted to the U.S. Social Security Administration; Pew, R. W., Miller, D..C., Grignetti, M. C., Massey, L. D. & Feurzeig, W. (1978). *Human factors test and evaluation facility: Interim report for the period 2 March 1978-30 November 1978.* BBN Report No. 3992. Submitted to the U.S. Social Security Administration.

116. Bowen, R. J., Feehrer, C. E., Nickerson, R. S., Spooner, R. L., & Triggs, T. J. (1971). *Army tactical intelligence: An examination of computer-based displays and a structured decision procedure as aids to the intelligence analyst.* BBN Report No. 2073. Submitted to U.S. Army Behavior and Systems Research Laboratory; Bowen, R. J., Feehrer, C. E., Nickerson, R. S., & Triggs, T. J. (1975). *Computer-based displays as aids in the production of army tactical intelligence.* Technical Paper 258, Army Research Institute for the Behavioral and Social Sciences; Bowen, R. J., Halpin, J. A., Long, C. A., Lukas, G., & Triggs, T. (1973). *Decision flow diagrams and data aggregation in army tactical intelligence.* BBN Report No. 2570. Submitted to U.S. Army Behavior and Systems Research Laboratory; Bowen, R. J., Nickerson, R. S., Spooner, R. L., & Triggs, T. J. (1970). *Army tactical intelligence: An investigation of current methods of production and possible computer-based innovations.* BBN Report No. 1945. Submitted to U.S. Army Behavior and Systems Research Laboratory.

117. Baron, S, Levison, W. H., Nickerson, R. S., Sutherland, W. R., & Thomas, E. L. (1970). *Development of a behavioral science programming language.* Report No. NAVTRADEVCEN 70-C-0046-1, U.S. Naval Training Equipment Center, Orlando, Florida; Nickerson, R. S., & Feehrer, C. E. (1975). *Decision making and training: A review of theoretical and empirical studies of decision making*

and their implications for the training of decision makers. NAVTRAEQUIPCEN 73-C-0128-1. U.S. Naval Training Equipment Center.

118. Nickerson, R. S., Adams, M. J., Pew, R. W., Swets, J. A., Fidell, S. A., Feehrer, C. E., Yntema, D. B., & Green, D. M. (1977). *The C3-system user. Vol. 1: A review of research on human performance as it relates to the design and operation of command, control and communication systems.* BBN Report No. 3459. Submitted to the Defense Advanced Research Projects Agency.

119. Feehrer, C., Miller, D., & Pew, R. W. (1981). *Evaluation of proposed control room improvements through analysis of critical operator decisions.* BBN Report No. 4394. Submitted to the Electric Power Research Institute.

120. Pew, R. W., & Feehrer, C. (1980). *Human factors evaluation of the EPRI safety panel/safety console designs.* BBN Report No. 4517. Submitted to the Electric Power Research Institute.

Chapter 9

Control Systems R&D at BBN

Sheldon Baron

The primary focus here is on BBN's theoretical and applied work on problems involving humans in the information processing and control loop of control systems. Discussions cover the evolution and outcome of key technical developments, people involved, and the role of BBN's environment — computational, organizational, and human.

9.1 Introduction

The problem of controlling some system or process so as to achieve a particular set of goals is ubiquitous, occurring in areas as diverse as biology and physiology, vehicle operation, robotics and the operation of other complex technological systems such as nuclear power plants and chemical and manufacturing processes. Although significant early developments in control systems design and implementation can be traced back a couple of centuries, a period of exponential growth in theory and practice was stimulated by the demands of WWII. This growth has continued unabated, fueled by the requirements for controlling increasingly complex technological systems and supported and abetted by advances in the mathematical theory of control and the development of more and more powerful computers and computational methods.

No single company, especially one the size of BBN,[1] could address the full range of control applications that emerged in the last half of the 20th century. There were, however, BBN activities and staff research interests that led to significant control system programs at the company. A broadening interest in human response to acoustical inputs fostered the development of a vigorous activity in experimental and engineering psychology which, in turn, led to a very long and productive R&D program in "human-in-the-loop control" (sometimes referred to as man-machine systems). In acoustics and related areas of physics, with improvements in sensors and in computational capability, active (rather than passive) control of the acoustic responses of systems became possible and desirable and, therefore, attracted significant efforts on the part of BBN staff. Later, as BBN became involved in computer network design, implementation and operation, it became necessary to address different kinds of control problems, for example those involving routing algorithms for packet switching or monitoring and control of large networks. In each of BBN's control system activities, the scope and duration of the efforts were dictated largely by the talents and interests of the staff and, very importantly, by an ability to find clients with financial resources and corresponding interests and needs.

This chapter will cover control systems work in just one of BBN's main areas of interest, namely, Information Sciences and Systems. In this area, emphasis will be on the work undertaken and accomplished within the Control Systems Department and its antecedent and descendent spin-off organizations. The primary focus will be on theoretical and applied work on problems that involve humans in the information processing and control loop. In addition, there will be brief discussions of some other interesting control applications projects that were important for the impacts they had on BBN and its customers. The discussions will cover the evolution and outcome of key technical developments, the people involved in the efforts and the role of BBN's environment (computational, organizational and human) on the work and vice-versa. Inasmuch as the chapter is something of a personal memoir as well as a historical review, a number of the author's personal observations and comments will also be included.

9.2 Human information processing and control

In this section, the history of BBN's work in control systems as it relates to "human-in-the-loop" control (i.e., human information processing, control and decision-making in dynamic, closed-loop environments[2]) is discussed. As part of this discussion, we will also include the work in developing decision aids for pilots and for advanced command and control applications as these can be viewed as natural evolutions of earlier control work. The work to be discussed covers the period from the 1960's through the 1990's, and was supported by a host of (NASA and DOD) projects.

Humans play a central role in monitoring and control of many such systems. Engineering descriptions (data and models) that describe the human's performance in terms that are commensurate with the descriptions of the inanimate portions of the system can be very useful for the analysis and design of these systems. Human controllers are complex control and information processing systems. It is generally acknowledged that they are adaptive and that their behavior in closed loop tasks is frequently time-varying, nonlinear and stochastic in nature. Accordingly, measuring, understanding and accounting for the human in these situations of continuous change and closed-loop feedback presents very challenging problems. When control theoretic approaches to modeling human performance were initiated in the 1950's and 60's, these problems were different in kind than those associated with the stimulus-response situations of interest in much of experimental psychology at that time.

Modeling human performance in manual control tasks

Research on the characterization of human performance in terms suitable for analyzing the manual control of continuous dynamic systems began in the 1950's. This work was rooted in servomechanism theory and in time series analysis techniques that were developed in the prior decade and were being adopted by control engineers in the design of automatic control systems. The analytical approaches of the time were mostly suitable for the study of linear, time-invariant dynamic systems, a class of systems that is amenable to both time- and frequency-domain analysis. In the span of over forty years since the beginning of work on analytical methods for treating manual control problems and systems, the objects of control have become increasingly complex and the technologies necessary for the automation of many control functions have advanced significantly. These developments have resulted in an evolution of the role of humans in controlling systems of interest from one of continuous manual control to one of supervisory control. As the various changes occurred, new methods of design, analysis and prediction of system performance were required and developed.

Early Efforts

BBN activities in the domain of human-in-the-loop control began, in earnest, in about 1960 and continue to this day. Almost all of this work was sponsored by various organizations in DOD and NASA. The initial work focused on human performance measurement and the development of mathematical models both for relatively simple manual control tasks and for monitoring behavior in more complex tasks. These efforts were related to, and sometimes drew on, those of BBN in psychology, but differed in that they focused on the development of techniques that enabled quantitative engineering analyses and predictions of performance of both the human and the human-machine system in closed-loop environments. Over the years, the efforts evolved and largely mirrored the advances in control and systems theory and practice, both in the problems considered and in the methods used to address them. The work also reflected strongly the changing interests and compositions of our clients and staff as well as the computational tools available at BBN and elsewhere.

One approach to modeling human behavior and performance, based on Information Theory, was pursued at BBN by John Senders. John came to BBN in 1963 from Minneapolis Honeywell, where he had led a Human Factors group. He was brought to BBN by J. C. R. Licklider and was put in charge of an Engineering Psychology department. John's principal research focus at BBN was on human information processing. In the beginning, he was mainly concerned with developing models for the visual sampling behavior of human pilots in cockpits with multiple displays and in predicting the impact of that behavior on overall performance. Such models would be useful for display panel design and other purposes. His initial modeling approach relied on Shannon's sampling theorem for data reconstruction.[3] He was able to obtain at least partial validation of his model using eye-movement data collected in flight tests at the Wright-Patterson Air Force Base in the early 50's. Additional experiments and studies were performed at BBN, through 1966, to fill gaps in data and to address some shortcomings in the model. Some of the experimental data was obtained from simulator studies using a Link trainer at BBN. New approaches to modeling pilot sampling strategies that potentially could address some of the problems with Senders' model, were pursued by others at BBN with varying degrees of success.[4]

About 1965, Senders began exploring the application of his ideas concerning human information processing to the investigation of automobile driving. This led to a report for the Bureau of Public Roads on the Attentional Demand of Automobile Driving. In addition, John developed a very innovative device for investigating a variety of aspects of the driving task; specifically, a helmet fitted with a mechanism that allowed the driver's vision to be occluded in a controllable fashion. The duration and frequency of occlusion could be set either by the investigator or voluntarily by the driver. Thus, if the investigator set the frequency, the driver could choose the vehicle's speed. Or, conversely, for a given constant speed, the driver could choose the frequency of occlusion to be at a level with which he felt comfortable. John showed that these variations could be related analytically to the rate of information presented by the roadway environment and, correspondingly, to the attentional demand of the situation.[5] This device was tested, for safety reasons, on a finished but unused stretch of a road that was under construction. It was tried later, with appropriate precautions, on a relatively busy street in Cambridge. John Senders' work at BBN has been cited many times in the psychology and human factors literature and, to this day, the notion of blanking displays to study attention is used as an experimental technique for studying visual attention issues. John left BBN in 1966 to teach at Brandeis University but remained a consultant to BBN for a year or two after that, mainly to finish his ongoing projects in automobile driving.

However, with John's departure and the completion of his projects, his approach to visual sampling research came to an end at BBN.

The work in continuous closed-loop control of dynamic systems by human operators (known in the field as manual control) that was based principally on control-theoretic ideas and techniques was started in 1960, stimulated by the arrival at BBN of Dr. Jerome (Jerry) Elkind . Jerry had completed a Sc.D at MIT. His thesis on the characteristics of simple manual control systems was an early and important contribution to research in manual control. His thesis advisor was J. C. R. Licklider, who was instrumental in bringing Jerry to BBN after Jerry had spent a couple of years at RCA. Jerry's work and that of other researchers in the late 50's, most notably that of Duane McRuer of Northrop Corp. and Ezra Krendel, at the Franklin Institute,[6,7] led to a class of models of human control behavior that employed quasi-linear describing function theory. The quasilinear model of the human operator consists of a describing function that accounts for the portion of the human controller's output that is linearly related to his input and a "remnant" term that represents the difference between the output of the describing function and that of the human controller. Simply put, the describing function portion of the quasilinear model for predicting human control behavior in a single-loop tracking or regulation task assumed that the behavior was such that closed-loop performance would approximate that of a "good" servomechanism. For more complex control problems, involving multi-input, multi-output configurations, a set of rules for choosing structures and parameters was developed by McRuer and his colleagues at STI, a company he founded to address pilot- vehicle control and dynamic analysis of aircraft and other vehicles. STI was a small, focused company and a (mostly friendly) competitor of BBN for government contract work in manual control throughout the time we pursued such work (until 1990's).

From 1960-1966, Elkind and his colleagues at BBN contributed in significant ways to the development of measurement techniques and experiments to support and extend quasilinear modeling. The experiments were conducted using analog computers to generate control system dynamics and related displays, as analog computation was the only viable method for real-time, person in the loop simulator studies at the time. However, data analysis was done on the PDP-1 and was performed using digital Fourier transform techniques. Some of the people who worked with Elkind on this effort were only at BBN a relatively short time but went on to distinguished academic careers elsewhere (notably, David Green and Lawrence Young[8]).

In 1964, Elkind hired William Levison, who had just completed a PhD in Electrical Engineering at MIT. Bill remained at BBN until 1997, all the while doing distinguished work in analysis and modeling of human performance. Bill had a tendency to work alone but later became part of the team that developed the OCM (see below). You never had to go far to find Bill-he was usually to be found in his office working very diligently. He was something of a skeptic and, most often, could be counted on to challenge new ideas or approaches. Although this could be deflating at times, it was also helpful as it forced one to defend one's ideas and thereby strengthen them or discard them if they didn't stand up. Bill was extremely well organized. He is reputed to have had every illustration and every paper he ever produced fully catalogued and numbered. This was no small accomplishment given the large number of publications he produced in his years at BBN. These characteristics would recede or disappear, however, at social gatherings when Bill would take his guitar in hand and sing along as he played. Then, his warmth and humor would emerge to surprise and delight us.

In the early 60's a major development was taking place in control theory spurred largely by three factors: 1)the need for precision and even optimization for tasks associated with the space program and with large-scale process control; 2)advances in digital computers which enabled direct digital control as well as the development of computational methods for solving complex control problems; and, 3) a trend toward treating control problems in abstract mathematical terms (i.e., a mathematical theory of control). These factors led to a new approach to control that focused on state- space problem descriptions in the time-domain, optimal and stochastic control problems, and digital solutions (often algorithmic) for these problems. These developments and the resulting solutions and techniques came to be called "modern control theory". At about the same time, other control theorists and practitioners were developing approaches for addressing problems in which systems changed, parametrically or structurally, in an unpredictable fashion over time. The developments in this area came under the general rubric of adaptive control.

Jerry Elkind saw in these developments in control theory the possibility for new approaches to analyzing the more complex, multi-variable human-machine control problems of interest to our clients, and for modeling human control in these problems. He proceeded to explore them in two contracts, with support from the Air Force and NASA. The Air Force contract explored the use of optimal control techniques for predicting control characteristics and display requirements in a helicopter hovering task whereas the NASA contract was a pilot study comparing the performance of human controllers with that of an optimal control system for a few situations. In addition, and more importantly, Jerry[9] began hiring people with the background necessary to address manual control from these perspectives. In 1965-66, Jerry Burchfiel and Duncan Miller (then PhD students with control backgrounds) were hired part-time, Peter Falb (a co-author, with Prof. Michael Athans, of a major text on optimal control) was brought in as a consultant, and Dave Kleinman who had just completed a PhD in control theory at MIT under Athans was hired as a full-time member of the staff.

Jerry's direct involvement in the manual control efforts, and in staffing for them, virtually ended in April of 1967 when he brought me to BBN to head a newly constituted Control Systems Department which essentially replaced the Engineering Psychology Department.[10] I came to BBN from the NASA Electronics Research Center (ERC) where I had led a group in Control Theory/Systems. I had completed my PhD in Applied Math at Harvard in 1966. My studies there included courses in modern control theory and a thesis in which I applied that theory to problems in differential games. I also had extensive prior experience at NASA Langley Research Center (LRC) where I had worked in aircraft stability and control, pilot-vehicle control, and real-time simulation, all of which would prove helpful in the manual control work being pursued by BBN. My educational background also included a BS & MA in Physics; this part of my background allowed me to appreciate those areas of BBN's research and development and influenced some of the later directions and activities that were initiated in the organizations for which I had responsibility.

The new Control Systems department included Bill Levison, Dave Kleinman and Jane ("Jinx") Ward (a research assistant to John Senders) as full-time staff, Duncan Miller on a part-time basis and Peter Falb and John Senders as consultants. The goals for the department were to continue and advance the work in manual control and to pursue other research avenues in, or related to, the control area. In the remainder of this section, and in the section that follows, we will discuss some of the accomplishments that were made in the pursuit of these objectives and we'll see how the work evolved with changes in technology, personnel and environment.

Before discussing the main thrusts of the efforts in manual control and their successors, a brief (somewhat personal) digression is useful. Shortly after I joined BBN, I began collaborating with Ray Nickerson and others on a study of vehicle rear lighting for the National Highway Safety Bureau. My part of the study involved developing a mathematical model of car following in a "string" of vehicles so as to investigate what information was required for safe car-following in the presence of disturbances in the string. In this analysis, the drivers were modeled as "ideal' or "optimal" controllers that were constrained by human limitations on observation and by an irreducible reaction time delay that was consistent with psychological data on human reaction times. The control "laws" were computed to maintain vehicle separations in an "optimal" manner. The vehicle string incorporating the driver models was then simulated on an analog computer and performance was evaluated for different assumptions concerning the information available to the drivers. This analysis demonstrated the utility of information concerning relative ("closing") velocity and acceleration in maintaining safe separations. It also showed the value of having such information for vehicles ahead of the one directly in front of one's own vehicle. These results helped provide support for the conclusions and recommendations that were made concerning rear lighting systems.

The work on rear lighting gave me an early opportunity to collaborate with Ray and others in the psychology department. It confirmed for me the value to our work in human operator modeling of interaction with BBN psychologists, despite any differences in approach or perspective. It may seem obvious that modeling human behavior requires such collaboration. However, most researchers that were developing engineering models for human-machine systems at the time were either not particularly concerned with input from the psychologists or did not have the opportunity to interact with quality researchers in engineering and psychology situated "right down the hall" from them . Fostering this interaction between modelers and psychologists at BBN became an important goal for me in both my research and management activities; I'm happy to say that the interaction persisted, and even expanded, over time.[11]

The Optimal Control Model

Upon my arrival at BBN, I began working with Dave Kleinman on the optimal control approach to modeling the human operator engaged in control tasks. In a few months, Bill Levison was working with us to form the team that was responsible, ultimately, for developing what we called the Optimal Control Model (OCM) for the human operator.[12] Within this team, Dave and I were largely responsible for the control theoretical ideas, developments and interpretations. Dave was also responsible for algorithm developments and software implementations of the optimal control solutions that were needed to yield numerical results from the model. Lastly, Bill had major responsibility for experimental work and for models for remnant and attention-sharing. However, this was truly a team effort with significant interactions and contributions across the team throughout the development.

By 1971, the model structure and parametrization necessary for application of the OCM were in place and we had obtained significant experimental validations of it. This model had emerged from several theoretical and experimental studies conducted in 1967-70.[13,14, 15,16,17] Very briefly, the OCM was based on the fundamental assumption that the well motivated, well-trained human operator will act in a near optimal manner subject to the operator's inherent limitations in processing information, executing actions and developing an accurate understanding of the task. This assumption was not new either in manual control or traditional human engineering studies. Without getting into details, what was novel about our approach were the methods used to represent delays and randomness in human observations and responses directly and the inclusion of those representations in the formulation of a stochastic optimal control problem, the solution of which was predictions of closed-loop system performance and human control response. The problem itself was a variant of the well-known Linear, Quadratic, Gaussian (LQG) of modern control theory and its solution required some modifications of the LQG solution because of the nature of the human limitations. A key aspect of the solution was that the emergent model for the human consisted of elements that compensated optimally for randomness and delays in human information processing which were separate from those elements that related directly to optimizing the control objectives.

The formulation of the problem and the resulting structure of the OCM had several important consequences. First, although the model was formulated in the time-domain, under appropriate assumptions of stationarity, a *prediction* of the human's describing function and remnant in the frequency domain could be made. Thus, it was possible to estimate the parameters representing human limitations and to validate the model against the detailed describing function and remnant data available from well-designed and controlled experiments in a class of simple control tasks. This was done and the results were rather remarkable in that very accurate predictions of closed-loop performance and human describing functions and remnant spectra were obtained. for a disparate set of control tasks. Moreover, the parameters used to represent human limitations were essentially fixed in obtaining these results. Second, because the model was formulated generally in state space and was, at root, a time-domain model it could be extended relatively gracefully to multi-input, multi-output systems and to non-stationary systems. Third, the separation of the information processing elements from the continuous control elements would eventually allow us to extend the approach to problems involving discrete decisions and supervisory control in dynamic environments. Finally, the general analytic expressions for the solution, though requiring algorithmic solution for specific cases, made computational solutions straightforward once the programs were implemented. These solutions provided predictions of the full statistics of response in a single "run". In that respect they were very efficient computationally.

The program implementation of the OCM was done on a time-shared computer (initially an SDS 940 and then the PDP10). It was an interactive program in which system and human inputs were initially made in a question/answer format. This was extremely useful for early investigations of and with the model. After a while as we became more familiar with the manner in which the model was being used, the Q/A format became tedious and we added the capability of creating an input file that was easily modified by an experienced user. I am convinced that because of the time-sharing environment, and our particular implementation, we made much faster and more efficient progress than would have been possible in a (then) more traditional batch computer environment.

After the initial validation of the model, and the presentation and publication of these results, there was a substantial spurt in activity involving the OCM. The largest share of the activity took place at BBN but, with time, it also extended to other organizations both commercial and academic and nationally and internationally. Despite a number of other attempts using different approaches, it is fair to say that by the mid 70's the OCM and the quasi-linear modeling being done by STI and its adherents were the dominant approaches for analyzing manual control problems.

The work in manual control at BBN continued very actively throughout the 70's and into the early 80's. It involved utilizing the OCM to investigate a diverse set new applications and problems, some of which required theoretical extensions of the OCM as well as new software implementations. It also involved a broader client base and a changing cadre of contributors. Some of the interesting new applications areas included: developing control and display requirements for aircraft approach to landing, both on land and at sea ; predicting human performance in AAA (anti-aircraft artillery) tracking systems; determining engineering requirements for flight simulators ; and, analyzing the effects of environmental stresses (e.g. vibration) on human control strategies and performance. These problems required using the model for analysis, prediction, diagnosis and even for experimental planning.

The fact that the OCM could be applied to this wide range of applications was a testimony to the soundness of the underlying approach and model structure. As noted, extensions of the model were sometimes needed to treat new or more complex situations and these, in turn, required new or modified software implementations. For example, the approach to landing problems involved consideration of time-varying system dynamics and input disturbances that were deterministic but unpredictable for the pilot (e.g., wind shear). The AAA tracking problem involved an input (a target aircraft fly-by) that was quasi-predictable once the speed and direction of the aircraft was estimated. And, the analysis of engineering requirements for simulators required modeling an expanded set of sensory cues (motion cues, outside the cockpit visual cues and tactile cues) and determining through experiments the parameter values needed for the model.

The staff involved in manual control research at BBN underwent a number of significant changes during this period. The team that developed the OCM suffered a significant loss when Dave Kleinman left in 1972 to start a regional office for Systems Control, Inc., a California company devoted to control applications. Dave was a key member of our team and also would be competing with us for work with the OCM. This turned out to be less of a problem than anticipated partly because, before too long, Dave decided to move on to an academic position with the University of Connecticut. With that move, he became a consultant to BBN in which role he not only helped us with our research but also provided us access to some of his high quality graduate students. At the University, Dave focused on modeling dynamic decision making and later team decision making. Although a theorist by nature and training, he remained true to the need for empirical verification that he was exposed to in his years at BBN. After many years at UCONN, Dave moved to a professorship at the Naval Postgraduate School where he continued to make valuable contributions to the field.

However, Bill Levison continued his full-time commitment to research in manual control throughout the 70's and the 80's. Much of his work was devoted to studying the effect of various physiological stresses on human performance using the OCM and associated methods of analysis. Bill was ultimately promoted to the position of Division Scientist. He left BBN in 1997 but continued to work on driver-vehicle modeling as an independent consultant for some time. During this period, Jeff Berliner, Greg Zacharias, Ramal Muralidharan, Roy Lancraft and Alper Caglayan were added to the staff and contributed to the human performance modeling efforts.

Jeff Berliner had a PhD in Electrical Engineering from MIT with a strong background in psychophysics. He assumed a major responsibility for development of the new software implementations of the model that were used internally and/or delivered to clients. Later, Jeff began working with the sensor signal-processing group within the department (see section on Multi-Target Tracking) where he made significant contributions to the projects in that group and its and organizational offshoots. Jeff eventually was promoted to the position of Division Scientist.

Dr. Greg Zacharias also joined us from MIT. Greg's PhD thesis was concerned with the mathematical modeling of the physiology of human sensory functions, particularly those associated with motion sensing in control tasks. His thesis advisor was the aforementioned Larry Young. He made immediate and important contributions to our work in several areas involving human-in-the-loop control, especially in the sensory perception areas and, later, in supervisory control.

Ramal Muralhidaran had a strong background in decision analysis and had a PhD from Harvard where his thesis advisor was Larry Ho (who had also been my thesis advisor). Ramal worked on adding models for discrete decision-making to the control models and on software implementations of the evolving models.

Roy Lancraft was a former student of Dave Kleinman at UCONN and was therefore familiar with the work in manual control. He worked on a number of different projects in that area. He also worked with Alper Caglyan, who came to us from NASA-Langley Research Center, on automatic failure detection systems for flight control. Alper had a Ph. D. from Virginia Polytechnic Institute and was highly recommended to me by former colleagues of mine at Langley. He had a strong theoretical background in control theory and experience in flight control problems. He made contributions to some theoretical problems required in extending the OCM in addition to obtaining support for his own projects on failure detection methods.

On a personal level, in July of 1971 I was elected to the position of Principal Scientist, the highest technical position in BBN at the time, but retained my position as Manager of the Control Systems Department. In 1975, I went on a six-month sabbatical to the Netherlands where I divided my time between the National Aerospace Laboratory NLR in Amsterdam and the Technical University at Delft. While there, I was able to devote much more concentrated time to research topics relevant to human information processing and developed a proposal, later funded by NASA, to apply the models to analyzing requirements for flight simulation. I spent a significant portion of the time giving lectures on modeling human performance and in working with Dutch colleagues on some of their related research. These efforts helped to advance the acceptance and use of the OCM and its later derivatives overseas. Over time, my role in manual control research diminished gradually as I pursued new areas of research. As some of these bore fruit (especially those in the signal and information processing area, see below), the Control Systems Department grew and much more of my time became devoted to management activities.

Extending the OCM to Modeling Supervisory Control

Towards the end of the decade of the 70's, it was becoming clear that the expanding complexity of the objects of control accompanied by the increasing need for, and capability to provide, automation, was altering the role that humans would play in the control of such systems. Thus, the human controller's task would become less and less one of continuous control and, more and more, would involve monitoring, decision-making and other supervisory activities. Moreover, the operation and control of these systems would frequently involve a team of people. The need for models and software tools to support the analysis and design of such systems was the impetus for a new direction in our work on human control in the latter half of the seventies and the 80's. There was a similar directional emphasis occurring in a significant segment of the human engineering community.

In the beginning of our work on supervisory control we were able to take advantage of our prior work on the OCM. Recall that the OCM had separable sub-models for sensory perception, state estimation and state prediction. These sub-models, taken together, form a model for human information processing in a dynamically changing environment. The model represents the operator's (cognitive) ability to construct a set of expectancies concerning the system state from an understanding of the system and incomplete and imperfect knowledge of the moment-by-moment state of the system. This set of expectancies can be used as the basis for monitoring, decision-making or other tasks (e.g., for continuous control as in the OCM). This information-processing model proved to be robust and fairly general. It was empirically validated indirectly by the validation of the OCM and, more directly, by a number of experiments involving monitoring and decision-making conducted at BBN and other places.[18,19]

From 1978-1982, we developed supervisory control models for three fairly complex situations and implemented them to demonstrate their capability. The first, called DEMON, sponsored by the Air Force, was a decision, monitoring and discrete control model for analyzing en route control of remotely piloted vehicles. DEMON added a decision mechanism based on an Expected Net Gain computation and a discrete control maneuver to the basic information processing architecture of the OCM. The second, called PROCRU (Procedure-Oriented Crew Model) sponsored by NASA, was developed to analyze flight crew procedures and both continuous and discrete control actions in a commercial ILS approach-to-landing.[20] PROCRU involved adding mechanisms for deciding when to perform certain procedures that were similar to those used in DEMON plus models for crew communication. It involved both continuous and discrete tasks. It was, and remains, the most complex model to be developed and implemented, using the OCM information processing model. The third, also sponsored by the Air Force, was called AAACRU and modeled the commander/gunner crew in an AAA system. It was similar in a number of respects to PROCRU and had an added model for situation assessment by the commander. Based on these models, we proposed a general supervisory control model architecture[21] as well as a model for supervisory control of a nuclear power plant. Unfortunately, for a number of reasons these two more generic models were never actually implemented in software. It should be mentioned that these supervisory control models were, inherently, simulation models in that they generated time-histories for a particular set of initial conditions and a particular sample of the random variables.

Major support for work on the approach to supervisory control that incorporated the OCM sub-models ended around 1985 except for a couple of minor efforts including one to develop an object-oriented software implementation of PROCRU. Probably, the principal reasons for this were changes in BBN and client personnel and organizations and the emergence of artificial intelligence (AI) technologies and cognitive science models as potentially useful ways to address the problems and issues in system design and implementation in supervisory control situations. Also, sometimes for very slowly changing systems and long operating times, the differential equations representations underlying the OCM were computationally inefficient compared to other methods such as discrete event simulation.

The changes in BBN personnel and organization relevant to these efforts began in 1979. At that time, the Control Systems department had grown to about 50 people with only six or seven working in the human control area and only three of them full time (Bill Levison, Greg Zacharias and Ramal Muralhidarn). I was devoting much more time to management and other technical areas of the department. In 1979, I was promoted to Divisional Vice-President and Assistant Division Director for the Information Sciences Division. This position involved more, and broader, management responsibility. BBN policy required that I give up the position of Principal Scientist in order to accept this promotion. I retained the position of department manager of a Control Systems department at that time. However, the department was much smaller because, as part of the divisional organizational change, we formed the Sensor Signal Processing department with Dick Estrada and Tom Fortmann as Manager and Assistant Manager, respectively. This new department was concerned with a variety of signal and data management activities that had been initiated with the OCTOPUS contract (see below). In addition to my continued direct management of the Control System department, as Assistant Manager of the Division, I had oversight management responsibility for the new Sensor Signal Processing department as well as for the Speech department and the Training Systems department, two departments with which I had previously established working relationships.

For me, the added management responsibilities made it difficult to devote much time to working on, or marketing for, supervisory control. Nonetheless, we were able to continue our momentum in the area for about four more years thanks to significant theoretical contributions from Greg and Ramal and software development by Ramal and Roy Lancraft (who joined the department in the latter part of the 70's). Theoretical contributions from Alper Caglyan of the department and from BBN psychologists were also helpful in this regard.

In 1984, the Information Sciences and Computer Systems Divisions were merged into a single Division (the Computer and Information Sciences Division) with Frank Heart as Director and Ray as Deputy Director. My management responsibilities were expanded further to include the Experimental Psychology Department. Also, under Dave Walden's leadership, BBN Systems and Technologies was attempting to increase our business in advanced systems work, a move that would involve larger projects than those typically available for supervisory control research. These changes, and some changes in my own interests, meant I had even less time for work in human operator modeling. Then, in 1983, Greg and Alper departed BBN, to start their own company, Charles River Analytics, Inc., which exists to this day. Greg also went on to have a very distinguished career in human performance research and Alper has since led a couple of other start-ups and has developed a recognized reputation in the area of intelligent agents. All this left us with less than a critical mass for pursuing and expanding our control-theoretic approach to modeling supervisory control. The Control Systems department was dissolved shortly thereafter with its remaining members assigned to other groups. In particular, Bill Levison joined the Experimental Psychology Department.

Use of artificial intelligence (AI) techniques in modeling supervisory control

One of the first things that motivated us to apply AI was that the procedural activities in PROCRU were represented as SITUATION-ACTION pairs (i.e., they were rule-based). In the original PROCRU implementation, these were coded in FORTRAN. This was not efficient computationally and made making changes or adding or removing pairs tedious. More importantly, it highlighted the fact that, in future investigations, modeling the discrete decisions and responses of members of a crew invariably would involve expressions of this form. On the other hand, the predicates for these pairs would often be based on continuous estimation of dynamically changing variables related to the state of the aircraft. In addition, when the conditions of several predicates were met, the decision as to which particular action should be taken required some means of priority evaluation that was consistent with the models of the limitations of human operators. Also, many aircraft states required continuous control. Thus, for our purposes, some blend of AI and control theory approaches was desirable. Ultimately, Ken Anderson, an AI software developer, produced an object-oriented model of the PROCRU models and scenario that ran on an AI platform and retained the appropriate continuous models. Regrettably, this was not taken any further as pursuit of PROCRU and similar models waned for the reasons mentioned above. However, a powerful approach to the closed-loop modeling and analysis of human-machine systems emerged. Though this approach was new it was infused with philosophies and elements that drew on the closed loop approach that preceded it.

Another motivation for our shift in approach to modeling supervisory control was the growing viability of applying the techniques and computational methods of AI to some of the problems of interest to us. Around 1980, AI and some of its associated computational techniques (especially rule-based systems and object-oriented programming) were attracting a surge of interest by system designers and (non-AI) researchers at BBN and elsewhere. At BBN this was particularly true in the Information Sciences Division which had a long-standing AI research activity (mostly in natural language and knowledge representation) and was home to a number of cognitive scientists (most notably Al Collins and Al Stevens). In 1980, Al Stevens, Bruce Roberts, et al had used object-oriented techniques in the "Steamer" simulation[22] to great advantage. In addition, starting in 1980 about 40 "Jericho" workstations for AI application and research were designed and built at BBN. With this environment, it was not surprising that many others in the Division were stimulated to look to AI for approaches to solving their increasingly complex problems.

For the staff concerned with human-in-the-loop control problems, this new interest was directed at two particular avenues of work. One was the development of workstation simulations of human-machine systems that incorporated cognitive models of the human and/or live operators. The second was the exploration of the use of AI to develop aircraft cockpit decision aids.

Workstation Simulation of Human-Machine Systems

In 1985, Kevin Corker joined the Experimental Psychology department from the technical staff at Jet Propulsion Lab where he had worked on human control of a teleoperator device. Before that he had worked for Hughes Aircraft on display and control analysis for advanced avionics systems. Kevin had a PhD from UCLA in Cognitive Psychology/Engineering Systems. His background and strong interest in the human-in-the-loop control problems that we were addressing made him an immediate and important contributor to our efforts.

Kevin took a leadership role in a number of research projects aimed at using human performance modeling integrated with various AI techniques to guide evaluation and design of commercial and military aircraft systems. The need for advanced, but relatively inexpensive, tools to explore the impact of introducing potential automation concepts into such systems was a strong motivating factor behind these efforts. In them, he developed techniques that combined object-oriented simulation/emulation approaches with cognitive, perceptual and psychomotor modeling in a workstation environment.

Around the time Kevin joined BBN, Dick Pew, the head of the Experimental Psychology Department, and a world-renowned figure in Human Factors, was awarded a contract from NASA Ames Research Center[23] (ARC) to develop and demonstrate an Advanced Task Analysis Methodology for the NASA-Army Aircrew/Aircraft Integration Program. This Center had sponsored many BBN research efforts in manual control, dating back to Jerry Elkind's work in the 1960's. They had also supported the development of the PROCRU model. Kevin was assigned the lead role for this new project. He and his colleagues developed a methodology that provided: a generalizable task analysis model; a modeling framework for aircrew interaction with helicopter systems; an index of critical workload and performance levels; explicit modeling of decision-making in mission execution; and a flexible mission analysis procedure and mission editor with suitable interface to models of vehicle and pilot behavior for the development of full system representation.

In another of these projects, for the Air Force Human Resources Laboratory (AFHRL), a simulation implementation of a tactical Command and Control system for Ground

Controlled Intercept was developed in a workstation environment that would support quantitative and qualitative predictions of human and system performance. In this simulation, real human operators could participate or some, or all of them, could be modeled using the representations Kevin had developed. The simulation incorporated speech recognition software developed by BBN and a commercially available speech synthesis product so that a live operator could interact verbally with simulated operators when required by the situation. This workstation simulation of crew operation of a complex system demonstrated the potential for providing a powerful means for evaluating prototype designs prior to committing to expensive hardware. The software development for the project was performed by Nichael Cramer a talented programmer with an excellent graphics capability. After some time, Nichael moved to a farm in Vermont but continued to work for BBN, with most of his development effort occurring off-site and his interaction via e-mail and phone.

The use of AI and discrete-event simulation led to another approach to human performance modeling at BBN in the form of the Operator Model Architecture (OMAR).While OMAR took on several different names at various points in its development, a consistent array of software building blocks have always made up its basic components. Several projects from different sponsors contributed to its development over the years, the principal long-term support came from the Air Force Research Laboratory (AFRL) and the NASA-ARC.

OMAR was first used as a human performance modeling system in SIMNET. In May of 1987, the tanks of the first platoon to cruise across the Ft. Knox simulated terrain each had a four-person crew made of OMAR models for a tank commander, gunner, loader, and driver. But before that, OMAR's first use had been as a discrete-event simulator to support course-of-action evaluation for the DARPA ALBM program.[24] Each of these efforts evolved through the collaboration of Stephen Downes-Martin, Glenn Abrett, and Steve Deutsch.

With the departure of Stephen Downes-Martin and Glen Abrett from BBN, Steve Deutsch led the further development of OMAR. Steve embarked on the development of human performance models in a series of tasks for AFRL and NASA ARC, modifying and extending the system to address various modeling issues along the way.

The AFHRL client for the workstation simulation effort described above continued to support various related efforts at BBN through the 90's (even after Kevin had left BBN in 1990 to join NASA-ARC). One of these efforts was in a research program to explore Agent-based Modeling and Behavior Representation (AMBR). BBN's role in this program was to provide a distributed simulation environment for AMBR experiments that would test and compare various modeling approaches proposed by other research institutions. The simulation environment called D-OMAR (for Distributed Operator Model Architecture) was based on the software described above and could be used to support both human participant trials and human performance model runs.

Cockpit Decision Aids and AI

The belief that AI was on the verge of moving beyond the research stage into one of providing useful approaches to practical problems was increasing in intensity around 1980. This was stimulated in large part by the fact that increased computational power was making various AI techniques feasible for real systems. Another very important factor was the emergence (some would say hype) of expert systems technology. It seemed that an AI "shingle" was being hung outside doors almost everywhere.

When the government agencies concerned with aviation-related research became interested in exploring AI it seemed clear to me that BBN was probably in a unique position to propose research in the area of applying AI to the development of cockpit decision aids. After all, we had extensive knowledge of aircraft display and control problems and of the strengths and limitations of the human pilot from our years of human-in-the-loop research. Importantly, BBN had a "real" in-house AI capability with a staff of significant size and experience. In short, we had the necessary interest, talent and expertise to perform this kind of research. Moreover, we were aware of many of the potential pitfalls that could stand in the way of real progress and application and were determined to avoid the hype that was appearing in the field.[25] And, finally, we had established relationships with government researchers who might be receptive to our ideas. These factors proved convincing to clients in the Air Force and in NASA.

Unfortunately, from the point of view of those of us in the control systems area, the AI department under Bill Woods was not particularly interested in nearer term applications of AI. They were focussed on knowledge representation and other longer term, basic research and had the funding to support those efforts. So, we could only rely on them for advice or occasional participation by an individual with a funding gap. This changed for the better for us when Walter Reitman assumed management of the AI department. Walter cooperated much more fully with us. He assumed significant responsibility for a particular project (see below) and he also assigned Robert (Bob) Schudy to support us fully in Avionics area. Bob remained in the AI department but reported directly to me. He became a major contributor to our work over the next several years. Walter also helped us by assisting in our interviews of potential additions to our staff. Eventually, we hired Bruce Wilcox and Richard Shu as members of the group working in this area, a group that I continued to manage. Also, Dick Pew, Carl Feehrer, Kevin Corker, Eva Hudlicka and others from the Experimental Psychology department worked hand-in-hand with us.

One of our early efforts for the Air Force was not solely or specifically an avionics application. This was a State-of-the Art Review of AI technologies areas performed for the Aerospace Medical Research Laboratory (AFAMRL) in support of their Automated Information Management Technology (AIM-Tech) Program. AMRL was a longstanding sponsor for much of our modeling work. The Aim-Tech program focused on three technical domains as areas for potential AI applications: systems design; pilot/aircrew automation; and command, control and communication. The eight AI technologies areas were: expert systems and knowledge engineering; natural language; knowledge representation; computer vision (image understanding); tutoring and training; planning and problem solving under real world conditions; AI tools and environments; and speech. In this study, Walter Reitman led the technical effort himself and he enlisted the support of BBN experts in each of the eight areas being reviewed. The review was designed to help the AMRL decide on an investment strategy for their AI efforts. The results were published in an AFAMRL technical report.[26]

In a project funded by the Air Force Wright Aeronautical Laboratories AART-1, the Artificial Intelligence Applications Study (AIAS),[27] led by Bob Schudy, we conducted a different kind of evaluation of the suitability of AI for avionics application. In particular, we examined the application of various techniques to tactical aircraft systems. BBN, with General Dynamics as a subcontractor, evaluated a wide range of potential applications using a numeric evidential evaluation technique developed in the study. The study incorporated expert opinion, provided by General Dynamics personnel, on four factors: operational; cost; technology; and risk. In addition to the study results, we developed a feasibility demonstration system for air threat assessment, which assessed the capability, opportunity, detectability, and intent of threats.

In the Avionics Expert Systems Definition Study,[28] also funded by the U.S. Air Force, BBN, again working with General Dynamics, used a systematic procedure to define an overall functional architecture for an advanced integrated intelligent avionics system. We then selected two systems (situation assessment and integrated display management) for detailed definition. We tested the system concepts using advanced prototyping and simulation techniques. After a very successful briefing of this work by Bob Schudy and Bruce Wilcox to the DARPA program manager for Pilot's Associate, we were funded to provide a demonstration in General Dynamics' dome simulation facilities in Fort Worth, TX. The demonstration consisted of an expert system to aid pilots in making decisions during frontal aspect intercepts working in real time with pilots in the loop. It was used as a (D0) demonstration for the Pilot's Associate Program. Our demonstration was a "working" real-time prototype (albeit of limited scope) and not a slide presentation like many AI "demonstrations" of the time.

Given our respective backgrounds and after the delivery of the D0 demonstration, it appeared that the BBN and General Dynamics team was the favorite to win a contract in response to a DARPA RFP (request for proposal) for a program to develop a Pilot's Associate. Besides our detailed technical proposal our team proposed an innovative contracting scheme. BBN would be prime contractor for the first 2 or 3 years when the program would be largely an AI research program and then, when evaluations in advanced flight simulators and considerations of transition to real avionics systems became paramount, the role of prime contractor would be assumed by General Dynamics. It was a major disappointment when we did not win one of the two contracts that were awarded. Although prior to the issuance of the RFP we were told that cost-sharing would not be a factor in the awards, each of the two winning bids provided millions of dollars in cost-sharing whereas our proposal contained virtually no cost sharing. As for our contracting scheme, the government found it imaginative and desirable but unworkable from their viewpoint because they would have to write a new contract if the primes were switched in the middle of the effort. Although this experience with the Pilot's Associate program was most disheartening, we learned a great deal in the process. In addition, we were spared, somewhat fortunately, from being involved in what turned out to be an extremely difficult program to execute successfully. Thus, we eventually were in a better position to apply our talent and resources to other AI avionics research and to Advanced Command and Control activities.

About the same time we were approaching the Air Force we also began discussions with NASA that led to a number of research projects. The genesis of these efforts is interesting and illustrative of how BBN sometimes received its research funding. A couple of years before the AI discussions began I gave an hour-long presentation on PROCRU to staff at the NASA-LRC. Kathy Abbott, a young research scientist in the audience contacted me sometime in the next month to see if I would be interested in funding for additional development and application of the model. Naturally, I was and we began some discussions to pursue the possibility. I discovered that Kathy had a background in Computer Science and a strong interest in human factors. So, in our discussions, I also indicated the directions we were examining in the use of AI for cockpit aiding. Ultimately, Kathy was unable to find the funding for PROCRU but did find support for an investigation of the application of AI to the development of Intelligent Aids for flight crew tasks in commercial transports. She became a group leader for this work at NASA-LRC and within a couple of years obtained her PhD in Computer Science specializing in AI. In a series of contracts that extended over the next seven or eight years, BBN conducted funded research for this group and supported their research efforts in AI and human factors.

As with some of our Air Force work, the Intelligent Aids study for NASA surveyed the potential applicability of various AI techniques for cockpit aiding. However, there were significant differences between studies. In this work, the focus was on commercial transports whereas in most of the Air Force work the concentration was on tactical fighter aircraft. The analysis and categorization of crew tasks appropriate for aiding also proceeded in a different fashion although all our studies relied on input from relevant pilot subjects. In this effort, there was significantly more attention paid to analyzing and annunciating the human factors issues associated with providing this kind of aiding to flight crews than in our other studies. Once the desirable areas and types of aids were identified and their respective values considered, we examined the state of AI as it pertained to implementing them. This allowed us to identify for NASA high-leverage areas of AI research as it related to cockpit aiding in commercial transports. Carl Feehrer, Bob Schudy and I were the principal BBN participants in this effort.

The above study effort identified the desirability and importance of having intelligent interfaces capable of managing the information presented to the crew in efficient and effective ways. Such interfaces would facilitate communication between the crew and on-board intelligent systems. We then studied what would be needed in the way of problem-solving capabilities to achieve such an intelligent aid. A model for crew information processing was developed that could be used to guide the development of both an intelligent interface and the underlying information processing for an intelligent aiding system. Finally, a prototype of such an aiding system was developed. The prototype, the Recovery Recommendation System (RECORS) was designed to operate in conjunction with FAULT-FINDER, a system for fault diagnosis in the case of engine failures that was designed by Kathy and other NASA personnel.

In 1988, we won a competitive procurement for a large (by BBN standards) delivery order contract from NASA for their Advanced Aviation Automation contract. This was a significant award for us and a little bit of the story surrounding it is interesting. We had been discussing possibilities for future work with NASA-LRC but were somewhat dismayed by the form of the contract described in the RFP. This type of procurement was not generally favorable for BBN because it was often decided on the basis of labor rates. It also appeared like the kind of contract that a large commercial aircraft company might desire enough to subsidize (the Pilot's Associate awards were still prominent in our memory). Nonetheless, we persevered in the belief (hope?) that our approach to the technical problem posed in the RFP plus our experience and prior performance would be sufficient to carry the day.

We prepared our proposal on early Macintosh machines connected on a local area net. This was the first time we had done so, having prepared all our previous proposals either on a typewriter or using a BBN developed word processor (SCRIBE) running on time-shared computer. Preparation, though intense, went smoothly until the final integration and printing of the various individual contributions. This process was excruciatingly slow, especially the production of the graphics, and it took the entire evening to produce one copy. At 7AM, my secretary Joan Groleau and I were making the necessary copies of the proposal on our fastest copying machine. We finished just in time for her to get on a plane and deliver them to NASA on time.

Under this contract, we performed a number of studies for NASA-LRC. The earliest studies were focused on AI but after a few years the emphasis of the effort shifted more toward human factors. Throughout the contract we placed a BBN staff member on-site at LRC with the individuals assigned reflecting the changed emphasis. In the first couple of years, it was Tom Reinhardt an experienced AI software developer. In the last years, Dr. William Rogers, a psychologist with strong experience in human factors engineering, was on-site working closely with LRC staff. The human factors work covered several subject areas related to Aviation Automation. Among others, these efforts included a laboratory investigation of pilot response to warnings, studies of situation awareness, and development of a principled approach to allocations of flight deck function in high-speed civil transportation.

In the AI area, our work for NASA focused on developing concepts for producing human-centered functional decompositions in the commercial flight domain and implementing those concepts in software. A functional decomposition is an analysis of the goals, functions, procedures and their interrelationships, as they relate to a set of general mission requirements. The principal goal for the software was to support the development of intelligent aiding systems, in part by developing means to explore methods for assessing pilot intent. The software was developed principally by Tom Reinhardt using a set of AI and simulation languages developed by BBN for other projects. This work turned out to be part of the new thread in human performance modeling at BBN that relied heavily on discrete event simulation and was continued in the OMAR related efforts described earlier.

9.3 Advanced command and control systems

In 1986, the AI department under the lead of Al Stevens and Ed Walker submitted two proposals to DARPA to develop knowledge-based decision-support systems as part of the Strategic Computing Program. BBN was the prime contractor for one called the Capabilities Assessment Expert System (CASES) with Advanced Decision Systems (ADS) and Grumman Data Systems (GDS) as subcontractors. In the other proposal, for the Air Land Battle Management (ALBM) system, BBN was a subcontractor to Lockheed Corp. These two proposals were successful but by the time they were awarded (or shortly thereafter) there had been an organizational change that shifted executive responsibility for the programs from Al Stevens to me (in parallel with a shift in executive responsibility for SIMNET from me to Al Stevens). The two programs were placed in a new department called Advanced Command and Control and I assumed the role of Acting Manager.

Fred Kulik, a retired Army Colonel, was the ALBM Program Manager. Dr. Fred Seibel, an experienced research scientist with a background in AI, served as the technical lead. This did not turn out to be a successful program for BBN, except for the development of some software tools that proved useful later. Our participation in the program lasted about two years. In our view, the lack of success largely resulted from a significant cultural mismatch between our staff and that of Lockheed. On the other hand, CASES was quite a successful program and, ultimately, helped launch a major activity in development of decision support systems for Command and Control. Over the next ten or twelve years this activity had several critical technical and operational successes. In the process, the Advanced Command and Control Department grew to over 100 staff members from an initial complement of fewer than ten.[29]

There were some aspects of this Department that were a departure from the traditional practices within the research and development parts of BBN. These were driven by the nature of the business of developing military systems. For one, we found that significant amounts of development had to be performed on location at the relevant government sites. This was often a contract requirement and it was useful because being close to the users was extremely important for development. But being on-site had its drawbacks as well. Development on-site was hampered by limitations on facilities, less access to the broad range of talents at the home office and, finally, a requirement to respond to the client daily. Eventually, we found it desirable to open a number of offices to serve the various military clients and, as a result, the department became considerably distributed. Thus, there were department members in the BBN offices in Cambridge, Washington and San Diego, and in offices we opened to support specific clients and programs in Hawaii, St. Louis and Norfolk. Another major departure from the usual BBN model was the hiring of a significant number of individuals with extensive military backgrounds. This was motivated by the necessity for having staff with a thorough understanding of the military problems being addressed, from both a user and institutional standpoint. These hires were not marketers or direct business developers- they were, instead, program managers and/or an integral part of the system development teams.

There were many people who were critical to the success of this venture, too many to mention all of them. First, it should be mentioned that there was extensive cross-departmental work in the area. Major contributions were made by Ed Walker and people in his AI department, and by staff working in the Distributed Computing department. Some of the principal contributors from these departments eventually transferred into the Advanced Command and Control Department. Notable among these transfers were Mike Dean, John Gunshenan and Gerard Donlon. Within the department, John Flynn and Ted Kral played the key management and business development roles.

John Flynn was a retired Navy Captain with combat and command experience. He was an Academy graduate with an MS in Computer Science from the Naval Postgraduate school. He had been hired by BBN to work in business development in 1986 after having served as a DARPA program manager. He had been a champion fencer, was an avid tennis player and sang competitively with a barbershop quartet. He transferred in to the Advanced Command and Control Department soon after initiation of the CASES program to become its Program Manager. After we hired Ted Kral and as the activity became more established around CASES and related projects and opportunities, John became the Manager of the Department with Ted as his Assistant Manager. John worked in the Washington office while Ted was in charge of San Diego operations.

Ted Kral was a retired Navy Lt. Commander with combat experience. He, too, was an Academy graduate and had a MS in computer science from Carnegie Mellon where his advisor was a leading figure in AI research. Ted had been DARPA Program Manager for CASES, which was how I came to know him. In that relationship, I came to respect his technical ability and his ability to set demanding goals for the program that were risky but had a reasonable chance of being achieved. He was fair and helpful. We hired Ted after we received the necessary assurances that his prior role posed no legal or ethical difficulties. However, to avoid even the appearance of a conflict, Ted did not participate directly in the CASES program for several years. I have never known a person who put more of himself into his work than Ted Kral. His talents and drive, along with his abilities to conceive and obtain projects and build and organize staff to execute them, were the major factors responsible for the growth of the department. After a year or two, with most of the departmental growth occurring in the San Diego office, Ted was made the Manager of the Department (and, eventually, a Vice-President of BBN Technologies). John Flynn continued to report directly to me as part of the business development activity for the Division but continued his very close association with Ted and his department.

As the Advanced Command and Control Department grew it became involved in many projects and developments. Here, we will discuss four of the early efforts that are both interesting and illustrative of the work and that also had major impacts for BBN and our clients.

CASES

It was late August of 1987 when we received notice that we had won the competition for CASES, more than a year after the proposal had been submitted. At that time we were told we would have to attend the contract kick-off meeting three days later at CINCPACFLT in Hawaii, the intended prime site for CASES development and installation. At that meeting, we were informed that the program was on shaky ground and that a prototype demonstration was needed in ten weeks in order to save the program. A team was put together to develop the prototype drawing on staff from BBN and its subcontractors Advancd Decision Systems and Grumman Data Systems.

John Gunshenan of BBN was chosen to lead the rapid prototyping technical effort required to produce the demonstration. John was a young computer hacker with great energy and skill. He had an extensive knowledge of the "off-the-shelf" technologies available within BBN in Cambridge and of members of the staff who might provide assistance to the effort. And, frankly, John was available, unmarried and willing to be uprooted to Hawaii for the next three months of intensive effort. His personal traits and knowledge stood him in good stead and he proved to be an excellent choice to lead the initial development.

A successful prototype was developed a couple of weeks before the deadline. This was accomplished thanks to a Herculean effort and, in very large measure, to the availability of a BBN software system, called CRONUS, that supported distributed computing. Because of CRONUS, the development team was able to integrate various warfare models and other elements that already existed on different computers with different operating systems with new code developed for the prototype. The resulting system was then demonstrated with a graphical interface running on a workstation. After the successful demonstration, I received an interesting and gratifying phone call from a senior member of DARPA. In the call, he told me he understood and appreciated the "blood on the floor" that was required to achieve what we had, and offered thanks on behalf of DARPA, the Navy and the country.

Once the prototype was finished, CASES development proceeded. This development took place largely on-site with BBN and its subcontractors providing both on-site and home-office personnel to support the effort. For BBN, the early on-site development team was John Gunshenan, Jim Chatigny and Jack Margerison. All three relocated to Hawaii to perform the work. Jim and Jack transferred into the department from the Physics Division. In the Cambridge office, the major contributor was Mike Dean. Later, as CASES evolved and matured, many others on the team and in the government contributed greatly to development of the system.

The original concept for CASES was that, essentially, it would be an expert system that captured the knowledge of a (soon to retire) operations analyst at CINCPACFLT and would produce future "static" assessments. These assessments were produced annually. In them, the relative warfighting capabilities of the United States and a potential adversary, were evaluated using either real or notional forces. The assessments could then be used to identify and interpret trends that would provide the basis for developing operational plans and determining the resources required to execute those plans. Shortly into the development, the goals for CASES changed, in part because the prototype and other developments suggested that it would be possible to provide "dynamic" assessments, i.e., evaluations that could take place within hours and days rather than weeks and months.

Over time, given experience with the prototype and advances in technology, CASES evolved. The final operational prototype system delivered to the government had the following characteristics. It was designed to be used to support warfare planning and analysis for real-world contingency operations and for notional standing war plans. It operated in a distributed network environment, communicating over the network with a standard operational data base which contained friendly and enemy positional information, unit and weapon systems characteristics and current targeting data. Although there was some expert system code embedded that helped the analyst make decisions based on the results of the model runs, this was not the major function or part of the evolved prototype. Hence, CASES as an acronym was modified to stand for Capabilities Assessment Simulation and Evaluation System to more accurately reflect the evolved state of the prototype.

The top-level control software and the expert system software operated under Unix, using Motif/X-Windows for all operator interactions with the system. All of the analytical models that were integrated into CASES operated on the same hardware as the top-level control software, allowing the system to be operated on a single Unix based system. However, each of the associated analytical warfare models could also operate on any hardware and software environment that was best suited for each specific model. Peculiarities associated with different hardware/software systems that supported the models were transparent to the top-level control software. The Cronus distributed network support software tool automatically handled differences in operating systems and languages. The set of analytical models had been ported onto various MIMD, SIMD and vector parallel machines to speed up the model execution times. The distributed design of CASES allowed for the use of these parallel computers, when they were available, in a way that was completely transparent to the user.[30]

The prototype CASES was completed at the end of 1990. The system had been used operationally by CINCPACFLT since the first incremental prototype release in 1988. In 1991, the Navy picked up sponsorship of the CASES program and BBN engaged in further development at NRAD in San Diego. Although CASES was not "transitioned" to the status of a formal operational system, the operational prototype was used extensively at various Navy sites for "real world" analyses and planning. CASES was also used as a "shadow" planning system for Desert Storm to check and augment the operational systems and plans.

DART

In 1990, BBN won several contracts sponsored by DARPA, Rome Air Development Center and USTRANSCOM under an initiative in Knowledge Based Planning and Scheduling. The initiative was called ARPI, which stood for ARPA Rome Laboratory Planning Initiative. The AI department headed by Ed Walker led the proposal efforts but they involved significant cross-departmental cooperation. These contracts constituted a comprehensive effort by DARPA to develop and deploy operational prototype systems and to create and employ a common prototyping environment. Work on this effort by BBN had already begun at TRANSCOM when Iraq invaded Kuwait, resulting in the initiation of Desert Shield. Based on some early operational successes with our ongoing work, the government sponsors requested that BBN (under the leadership of Ted Kral) fast track development of a proposed decision support system to help planners in scheduling and analysis for the movement of equipment and personnel to military operations. The operational need to expeditiously move forces from the United States and Europe to Saudi Arabia dictated compressing the 18-month scheduled development time. Ted Kral then led a 10-week on-site development effort involving staff from BBN and its subcontractors Ascent Technology, SRA and ISX Corporation. This prototype system called the Dynamic Analysis and Replanning Tool (DART) was deployed in eight weeks, about halfway through Desert Shield. The BBN staff involved in this intense and successful effort were Ted Kral, Huong Ton (HT), and Mike Dean[31] from the San Diego office; and, Dick Estrada, Jeff Berliner, Mike Thome and Gerard Donlon from the Cambridge office AI department.

DART was revolutionary for its time and in the context of the application. The system in use at the time ran on a mainframe and produced reams of printouts to be analyzed. DART was, instead, a GUI-based scheduler that allowed users to interact with the system and to run transportation models in minutes rather than in the hours or days it took on the system then in use. These improvements and others enabled planners using DART to consider more alternatives and, thereby, to develop a more realistic plan in much less time.

Dart was an extremely successful program. It was subsequently hardened and deployed to several operational sites. It became a key component of the DOD JOPES (Joint Operational Planning and Execution System). In addition, according to the Director of DARPA at the time, Victor Reis, DART paid back 30 years of investment in AI by DARPA in a matter of a few months. Largely for its work on DART, BBN was named DARPA contractor of the year. Finally, DART may be viewed as the precursor of a string of BBN efforts in Logistics planning.

Distributed collaborative planning

DART was initially developed as a stand-alone planning system. However, the logistic planning problem was inherently distributed, with many different commands involved in the process leading up to a full TPFDD (Time Phased Force Deployment Data) operational plan. This was true also of other command and control systems and people at BBN were seeking opportunities to develop distributed planning systems.

Around the same time as development of the DART system, BBN was also developing the secure DSI (Defense Simulation Internet) for another part of DARPA. DSI was originally envisioned as a wideband network to support distributed simulation systems. However, in the early phases of DSI development and deployment there were very limited distributed simulation applications available. The DSI developers were anxious to identify other software systems that might be used to test and showcase quickly the emerging DSI network. This situation resulted in a happy circumstance of a need for distributed system support for DART and a new wideband network with lots of available capacity. A key factor in the marriage of these technologies was that both systems were being developed by DARPA and the DARPA Program Managers were willing to cooperate.

The initial marriage of the DART logistics planning system and the DSI wideband network occurred in 1991 at the JOPES planning conference in Atlanta Georgia. BBN was responsible for putting together one of the major demonstrations at the JOPES conference and suggested to DARPA that this was a good opportunity to showcase both the DART and DSI capabilities. With the go-ahead from DARPA, BBN assembled and integrated a set of applications that the company had under development to provide TCP-IP based video teleconferencing and a shared map and interactive viewgraphs to support the collaborative use of the DART logistics planning system. Remote sites at the U.S. Pacific Command in Hawaii and the U.S. Transportation Command in St. Louis, MO were connected using the DSI with the demonstration site at the JOPES conference in Atlanta, GA. BBN's John Flynn coined the term Distributed Collaborative Planning (DCP) to categorize the integrated capabilities first displayed at the JOPES conference. Shortly thereafter, BBN applied the DCP concept to other command and control systems they were developing, including CASES and the Theater Analysis Replanning and Graphic Evaluation Tool (TARGET).

The successful demonstration of DCP by BBN at the 1991 JOPES conference was a watershed event in the history of military planning systems. From that point on stand-alone systems were considered obsolete and today the term Distributed Collaborative Planning is widely used within the DOD, and around the world, to describe many different distributed planning systems that now use the infrastructure of the Internet and World Wide Web.

TARGET

The Theater-level Analysis, Replanning and Graphical Execution Toolbox (TARGET) is a system developed by BBN initially under sponsorship of DARPA and Rome Laboratory. It demonstrated the applicability of various advanced technologies developed under the Knowledge Based-Planning and Scheduling Initiative. TARGET is an integrated set of tools developed to support the military planning process. It provides an advanced decision-support environment for the creation, storage and exchange of planning information.

Target also provides the capability to perform distributed collaborative planning. Integration of the planning activities is achieved via a common plan representation and an underlying commercial off-the-shelf object-oriented knowledge base. Wide-area packet-switched and ATM networks are used to provide the real-time communications among Target users.

Using TARGET, military planners could: display situation assessment information, provide updates to it from remote sites; develop and evaluate multiple courses of action collaboratively; provide a common view of planning information to other decision-aiding systems; and support execution via quick situational updates and rapid replanning.

TARGET was used as a component of the DARPA Joint Task Force ATD and the Advanced Joint Planning ACTD efforts.

9.4 Applications not focused on human control or decision support

Some attempts at diversification with modest success

There were several attempts made to expand the control systems work to areas other than human-in-the-loop control. These included the areas of robotic control, hierarchical control, control of large space structures, differential games, and failure detection in avionics systems. Often, an area was started with a speculative hire of a "hero" in a particular area that was generally deemed to be an important opportunity for future support. Examples of this type of hire were Dr. Tim Johnson from MIT who was an expert in control theory and focused on robotics and hierarchical control, Mark Balas who performed highly regarded research in the control of large space structures and the aforementioned Alper Caglayan who had established credentials in failure detection problems. These efforts produced high quality, innovative technical results and received outside contract support for a time. However, for a variety of reasons, none of them proved to be capable of obtaining the kind of longer term outside support that would be necessary for them to be self-sustaining and, thereby, successful in the BBN environment. As a result, the individuals mentioned above eventually left BBN to pursue their interest elsewhere. Nonetheless, we benefited from the intellectual stimulation their work provided by their interactions with the staff and through contributions they made to other projects while at BBN.

There was one effort to expand our control and information processing related activities that proved to be directly, and indirectly, instrumental in providing major successes for BBN in the long term. This effort involved the application of the modern filtering or estimation techniques (a.k.a., Kalman filtering) to multi-target tracking of undersea targets. We will discuss this application, its genesis and its ultimate impact in some detail.

Multi-target tracking and sensor signal processing

In 1973, Howie Briscoe of the Computer Systems Division had an opportunity to submit a proposal to DARPA in the area of undersea tracking. Howie had a strong background in signal processing systems. (For reasons I can't remember) I was asked for ideas for a technical approach to the problem. I proposed a new, high-risk approach to the problem that involved applying modern (Kalman) estimation techniques coupled with decision theory to perform multi-target tracking. I was very familiar with estimation theory, especially from our work on the OCM and its extensions, and was eager to apply these notions to other problems. I had been following the multi-target tracking literature where elements of the approach were being applied to ballistic missile tracking because of general interest and the possibility that the ideas being explored there might be of use in future applications we might encounter.

I proposed two "wrinkles" that set the approach apart from prior approaches to undersea tracking. First, the "targets" to be tracked included both targets of interest and ships that might otherwise be considered as noise. This would add considerable computational burden but, potentially, it could support better data association in the presence of significant "shipping noise" thereby providing much better performance in the tracking of the "true" targets. Second, the mathematical models required for the Kalman Filters (or estimators) were to be derived to the extent possible from knowledge concerning underwater acoustics and true target characteristics. This approach could also add to the computational load and complexity but having such models could improve filtering, classifications decisions and maneuver detection, all of which could enhance the detection and tracking of the targets of interest. I proposed drawing on experienced staff from our Physics Division to help provide the necessary models. Thus, the proposal was multi-disciplinary in nature, drew on obvious strengths of BBN and the approach was in the high risk, high reward category that fell within the DARPA mandate.

We were awarded a multi-year contract based on this proposal. Eventually, this project led to the development and implementation of the tracking system, which was given the name OCTOPUS. The main implementation of OCTOPUS was at the DARPA Acoustic Research Center[23] (ARC), where it contributed to the larger undersea signal processing and data analysis research being conducted.

However, the real impact and importance of this project for BBN was that it provided us the initial support necessary to hire two exceptional people, who, in turn, played the major roles in the development of substantial activities in sensor signal processing and in data analysis. The growth in these areas fueled the further hiring of extremely talented people thus providing the kind of multiplier effect that was so desirable in BBN's environment. And, later, several of the individuals whose careers at BBN started in connection with the expansion of these areas assumed positions of importance in BBN.

After some preliminary work on the project, it became clear we were under-staffed with respect to the modern control/estimation theory expertise that we needed and desired. We were very fortunate that Tom Fortmann became available at just about the right time and was able to join BBN in 1974. Tom had been teaching at the University of Newcastle, Australia. He had a BS in Physics and M.S. in Electrical Engineering from Stanford and a PhD in Electrical Engineering from MIT. His thesis advisor at MIT was Mike Athans, a leading control theorist of the times. In addition to his excellent technical skills, Tom was clearly a highly energetic, motivated, organized and dedicated individual who also possessed an outgoing personality. It wasn't too long before Tom became the lead technical person for developing both the theoretical basis and the implementation of OCTOPUS. Along with leading the software development, Tom worked on establishing underlying theoretical bases for the tracker and, later, he collaborated with Yaakov Bar Shalom (a consultant on the project and a professor at the University of Connecticut) on a book on tracking and data association. Tom's leadership on the project had the ancillary benefit of allowing me to act largely in an advisory capacity on the project and freeing me to pursue my research in human control and other potential new areas for growth in control systems.

Then, in 1975 we were able to hire Dick Estrada to help support the project and to develop new business in digital signal processing for undersea applications. Dick had a PhD from the University of California at Berkeley with a control-related thesis in stability theory. After obtaining his degree, he worked at Bell Labs developing signal-processing algorithms for long range detection of ocean targets. His background fit in with the group and his experience was extremely important for the OCTOPUS project and any future work in the area. At the ARC, OCTOPUS was implemented using data provided by a signal processing algorithm (SIMS) developed by Dick (see Chapter 10). Dick had a keen and active sense of humor and appeared to be laid-back and somewhat disorganized, but these traits belied a powerful intellect, a strong work ethic, an ability to lead technical people and an entrepreneurial outlook. Subsequently, these talents helped him develop, obtain and lead a significant number of important research projects in the future.

Tom's Fortmann's career at BBN was long (24 years) and extremely productive. He developed an automated data analysis software capability with Steve Milligan (see Chapter 11) that led to a major government business area and, eventually, to a commercial business and a product called DataProbe. In 1990, he was elected an IEEE Fellow, cited for Technical Leadership in automated data analysis and multi-target tracking. Tom also had a very successful managerial career at BBN serving, successively, as: Assistant Department Manager of the Sensor Signal Processing Department; Department Manager of the Automated Systems Department; and Vice-President of BBN Systems and Technologies. As Manager of the Automated System Department he played a significant part in the development of the business at NUSC and in the activities of the Newport office where he helped to transition technology developed in Cambridge. The site manager for Newport at the time was Tad Elmer. He reported to Tom and benefited from Tom's support and guidance. Tad is now President of BBN Technologies.

Dick Estrada's career also had major impacts on BBN. He was a prime mover in developing the digital signal processing activities of BBN at the Acoustic Research Center. He was largely responsible for the growth of the area within the Control Systems Department. In 1979, when I became Assistant Division Director of the Information Sciences Division, Dick was appointed Department Manager of the Sensor Signal Processing Department, which by that time numbered about 40 employees. (Tom Fortmann was the Assistant Manager). He continued to develop programs that applied advanced computer technologies to undersea surveillance culminating in the ARIADNE program (see Chapter 10). In 1988 he was named a Vice-President of BBN Systems and Technologies. Dick played a major role in the FDS proposal effort and the early stages of the program. It is not an overstatement to say that without Dick's contributions in this area BBN would not have been in a position to command a major role in FDS. When the FDS contract was awarded to IBM and BBN, Dick's department was moved to the Physical Sciences Division. After about a year on that program Dick returned to the Information Sciences Division and worked in the AI department. He then played a principal role in developing the Logistics business for the company.

During the growth period of the 1970s when the sensor signal processing activities were still part of the Control Systems department there were a number of other staff additions that proved to be very important for BBN, indeed too many to mention all of them here. Foremost among these was Steve Milligan, who joined the Department in 1978. Steve has made enormous and diverse technical contributions to various efforts in his career at BBN. Without getting into details, suffice to say that Steve was named Principal Engineer, then Chief Scientist for Information Sciences and Technologies and is now BBN's Chief Scientist. Another key hire was Herb Gish, an expert in probability theory and its application in signal processing who joined the group in 1977. After contributing to many of the sensor signal processing projects, Herb transferred to the Speech Department where he has made significant contributions in language processing using statistical techniques. Based on his outstanding achievements in this area, Herb was promoted to Principal Scientist in 1997.

Finally, it is interesting to mention that as the activity grew and there was an increasing need for software developers, the group hired several undergraduates from local academic institutions for part-time and summer positions. It was not unusual for BBN to hire students to act as research assistants. But the scale of the hiring in the sensor signal processing group, the level of responsibility given to the individuals and their performance in meeting the responsibility was unusual. Many of these "kids" worked for BBN for several years and one, Ron Scott, earned a PhD in mathematics and became a full-time employee. Ron remained at BBN for several years, making significant contributions during his tenure. This experience served as a model a few years later when a similar approach was used to help meet the extensive software development needs for SIMNET.

9.5 Concluding remarks

In this chapter, we have examined the work in the control area at BBN over a period of approximately 40 years. In going through this history, the technical contributions, the people involved in the efforts, the interaction with other activities and technology at BBN and the impacts of, and on, various organizational changes on the activities have been discussed. It is fair to say that this area of work was not one that could be called a major thrust of BBN when compared to the larger areas of computer sciences and systems, networking or acoustics. Nonetheless, it is also valid to point out that the activity provided significant technical advances in the areas it pursued and was recognized internationally in its field, while, almost always, recovering its costs and generating a profit. Moreover, and perhaps more importantly, the activities also had significant impacts on BBN, of which four particularly noteworthy ones are summarized very briefly below.

First, the leading edge technical contributions in the field of human performance modeling added to BBN's reputation for technical excellence and achievement. It also lent additional credence and breadth to BBN's capability for addressing problems of human-machine and human-computer interaction that proved very useful in addressing systems problems.

Second, the activity spawned new, viable and substantial business areas in sensor signal processing, data analysis and command and control. The sensor signal processing activity ultimately led to major contracts in undersea surveillance, including the FDS contract. In the data analysis area, in addition to substantial amounts of significant contact work, a successful BBN data analysis product, DataProbe, was developed. In Command and Control, the activity ultimately grew to be one of BBN's largest departments winning and performing successfully many significant defense systems projects.

Third, the hiring and development of many individuals who went on to make major contributions to BBN, both technically and in management. Most prominently among them were individuals who attained the following positions: one Senior Vice President, five Vice-Presidents, one Chief Scientist and two Principal Scientists. These individuals, also had well-established technical reputations outside of BBN based on their efforts within the company, as do many others who participated actively in this activity. Several others, who contributed to BBN but chose to leave the company for other positions, went on to distinguished careers elsewhere and often maintained a fruitful connection to BBN.

Lastly, the control systems activity and its descendants demonstrated frequently the power of multi-disciplinary, inter-departmental work at BBN with projects at various times that involved participation with the activities and departments in Experimental Psychology, AI, Acoustics and Computer Systems.

These impacts are more remarkable when one considers that as a relatively small technical area, and one without obvious product potential, there was virtually no internal financial investment for the activity in the control area other than the department's overhead money for marketing and writing proposals. Inasmuch as this money directly impacted bidding rates, it had to be carefully monitored and limited to keep those rates reasonably competitive. This meant that in the BBN environment the sine qua non for a department was "billability," so that costs could be recovered and a modest profit made, even though many of the costs that were to be recovered were not under departmental control. As the company grew and investments were made in the more promising avenues for growth, bidding rates increased and BBN gained a reputation for being "expensive" relative to its competition. In the world of human performance modeling, where the competition was most often either a small business devoted largely to this technical area or an academic institution, this reputation and its underlying reality had significant effects on our competitive position. In the earliest days of the activity, it was possible to persist by selling our leading edge ideas to obtain sole-source contracts through unsolicited proposals. As the government procurement environment changed to emphasize competitive bidding, and as BBN grew so we were not eligible for small business set-asides or grants, the situation changed and became more tenuous.

So what was it about BBN that allowed us to sustain the control related activity in the field of human-in-the- loop control for almost 40 years and to obtain and keep a reputation and status as a major leader in the field. And, what were the keys, if any, that account for the other impacts the activity had on BBN's technical world and business. Of course, the achievements were driven principally by the people involved, that is, by their individual talents and drives. However, in the author's opinion, there is more to it than that and it has to do with the culture and environment at BBN. Specifically, I believe there were five other general factors that contributed significantly to the successes associated with the controls area, namely: the hiring practices that were part of the BBN culture; the technical environment; the support of management; the competitive environment; and the operating and organizational environment. I discuss these below.

The hiring practices at BBN remained rooted in the fundamental notion of hiring for excellence, as espoused by its founders. In the control systems area, we hired on the assumption that we would be prepared for the individual to spend his or her career at BBN. Accordingly, there were rigorous interviews by as many of the staff as possible to assure that the individual got a good feel for BBN and that we would be comfortable with him or her. On a technical level, we required that the individuals have state-of-the-art knowledge and a demonstrable level of performance. For new graduates, this was often evident from the level of academic achievement of the individual and from references of professors who were known to us personally. For more senior individuals, we looked for demonstrated job performance and for entrepreneurial skills and attitudes where appropriate. Of course, we made a few mistakes in hiring over the years but, on the whole, we had an excellent batting average. A small drawback of our hiring practice was that the highly qualified and achievement-oriented individuals we hired were often heavily recruited after they joined BBN, or they became interested in venturing off on their own. Hence, we lost some excellent people who left either to start their own businesses or because they received very appealing financial offers that we could not match. Almost always, relationships with individuals who left remained cordial and, sometimes, they returned to BBN at a later time or we found other ways to work together again.

The technical background at BBN was unusual, if not unique, in a number of ways, especially for an organization of its size. Much of this background is discussed in fair detail in other chapters of this issue. As we've seen in the account above, the control area was able to capitalize on BBN's strengths in a number of important ways. The diverse nature of the technical activities and the level of expertise of the staff engaged in each activity were a source of real strength. In particular, the psychologists and human factors staff were very important in supporting and grounding our research in human-in-the loop control. Research and staff in the computer and information sciences and systems areas were key contributors to our growth. Specifically, the world-class and pioneering work in AI, distributed computing and networking provided a very important impetus for our research efforts in human modeling and decision aiding and for advanced systems development in the Command and Control area. And, BBN's extensive knowledge and prominent position in undersea acoustics was an essential element for our entry and success in the sensor signal processing area. A second feature that was of great importance for us was the computational environment at BBN, which continuously and consistently evolved so as to be at the leading edge of the field. This environment allowed us to work at high levels of efficiency and provided the knowledge and stimulus for proposing systems that could capitalize on like advances. This technical environment was enriched and sustained by the fact that BBN had a technical "ladder" through which the research-oriented staff could experience professional growth and by the existence of a substantial Science Development Program that was was very helpful in attracting, developing and keeping high quality staff.

It was mentioned above that there was not much direct financial investment for the control area as it did not represent a major thrust for BBN. On the other hand, we had other kinds of management support for our efforts. This other support which was available to all at BBN included: strong support from the administrative departments in human resources, contracts and finance; state-of-the-art, or better, computer resources; and the backing of upper management to pursue our activities so long as they produced first-rate technical work and they were financially self-sufficient. This situation was typical of that afforded other research and development activities at BBN. I would add that, from my perspective, by and large, the management demonstrated an interest in, and respect for, the control-related activities in which we were engaged and this was important for staff morale.

The competitive environment presented some difficulties for the control work as noted above. We published the results of our research promptly, making it possible for others to use or capitalize on them to compete with us for new work. This added to the difficulty associated with often being unable to compete on price. However, the challenge that this environment presented had beneficial effects too. We had to be more innovative in our approach and we had to make the most of the advantages the BBN environment provided us. Although we would have preferred an environment in which price was less of a factor, on the whole, I believe the competition proved to be healthy in causing us to continuously push the state-of-the-art of our research and to find ways of providing added value to our clients. In the end, the unfavorable aspects of the competitive environment did not prevent us from succeeding and I believe the results were better than they would have been absent the competition.

The operating environment and organizational structure of BBN contributed in some subtle ways to the success of the control area. Departments operated within a divisional structure. From a top management viewpoint, divisions were the profit centers. However, within a division, separate accounting was kept for each department so that it had its own "P&L" statement. Although there were some negative impacts of this arrangement, there were also benefits, chief among them being an ability to smooth out temporary shortfalls in support in a single department thereby avoiding cuts in valued staff. This allowed the control area to survive some tough moments relatively unscathed (and, on occasion, to help other departments as well). Another significant advantage of the arrangement was that department heads had considerable autonomy and responsibility provided they managed their operations effectively. Also, it should be mentioned that all managers in the research and development divisions of BBN had strong technical backgrounds and this contributed to an atmosphere of heightened understanding and respect between managers and staff.

Thus, to summarize succinctly, a symbiotic mix of very talented technical and entrepreneurial individuals and an unusual and highly desirable culture and environment provided the basis for efforts that produced world class technical outputs and impacts on the company beyond what could have been expected given the size of the activity and the level of company investment.

Acknowledgments

John Flynn, Steve Deutsche, Ray Nickerson, Dick Pew, Dick Estrada, Bill Levison, Ted Kral, and Kevin Corker all provided helpful inputs to my memory and thereby to the preparation of the chapter. I also want to thank all my colleagues at BBN, technical, administrative and management, in my 31 years at the company that are the major reason that I have the material to write about. Finally, a special thanks to Dave Walden and Ray Nickerson for encouraging me to write the chapter from the personal people-oriented perspective.

Notes and References

1. During the time period covered in this chapter the name of BBN's research and development organization changed several times. To avoid confusion, I'll refer to the relevant organization by the generic name BBN.

2. These dynamic environments and their mathematical descriptions distinguish this work from more traditional areas of stimulus-response descriptions of psychology or the methods often employed in cognitive science.

3. John W. Senders, "The Human Operator as a Monitor and Controller of Multi-Degree of Freedom Systems," *IEEE Transactions on Human Factors in Electronics*, HFE-5, No. 1, September 1964, pp. 2–5.

4. John. W. Senders, Jerome I. Elkind, Mario C. Grignetti, and Richard Smallwood, "An Investigation of the Visual Sampling Behavior of Human Observers," NASA CR-434, April 1966.

5. It has been said that John came up with this idea while driving in a rain storm which caused his vision to be heavily obscured during periods when the wipers did not clear the glass.

6. Duane T. McRuer and Ezra S. Krendel, "Dynamic Response of Human Operators," Wright Air Development Center, Wright-Patterson AFB, Ohio, Report WADC TR56-524, October 1957.

7. Duane T. McRuer, Dunstan Graham, Ezra S. Krendel and W. Reisner, Jr. Human Pilot Dynamics in Compensatory Systems: Theory, Models and Experiments with Controlled-Element and Forcing Function Variations, AFFDL-TR-65-15, July 1965.

8. Larry Young became a Professor at MIT and was later selected for the astronaut program for the Shuttle program but, unfortunately, he never got to fly. Among Green's notable contributions were his collaborations with John Swets in the field of human signal detection theory.

9. By this time, Jerry was deeply involved in management of a division devoted to a range of human and computer problems so his personal involvement in manual control research was rapidly diminishing.

10. Editor's note: An Experimental Psychology Department, managed by Joe Markowitz, continued to exist.

11. My personal association with Ray and other BBN psychologists (particularly Dick Pew, John Swets and Carl Feehrer) also provided me with enormous intellectual stimulation as well as life-long friendships.

12. Given later developments and emphases, we probably would have chosen a different name for the model but by then the OCM designation was fairly firmly entrenched in the community.

13. Sheldon Baron and David L. Kleinman, "The Human as an Optimal Controller and Information Processor," NASA CR-1151, September 1968.

14. David L. Kleinman, Sheldon Baron and William H. Levison, "An Optimal Control Model of Human Response. Part 1: Theory and Validation," *Automatica*, Vol. 6, No. 3, May 1970, pp. 357–369.

15. Sheldon Baron, David L. Kleinman, and William H. Levison, "An Optimal Control Model of Human Response. Part 2: Prediction of Human Performance in a Complex Task," *Automatica*, Vol. 6, No. 3, May 1970, pp. 371–383.

16. William H. Levison, Sheldon Baron and David L. Kleinman, "A Model for Human Controller Remnant," *IEEE Transactions on Man-Machine Systems*, Vol. MMS-10, December 1969.

17. Sheldon Baron and William H. Levison, "An Optimal Control Methodology for Analyzing the Effects of Display Parameters on Performance and Workload in Manual Flight Control," IEEE Trans. On Systems, *Man and Cybernetics*, Vol. SMC-5, No. 4, July 1975, pp.423–430.

18. William H. Levison and Robert B. Tanner, "A Control Theory Model for Human Decision Making," NASA CR-1953, December 1971.

19. Sheldon Baron, Greg Zacharias, Ramal Muralidharan and Roy Lancraft, "PROCRU: A Model for Analyzing Flight Crew Procedures in Approach to Landing," NASA CR 152397, 1980.

20. Sheldon Baron, "A Control-Theoretic Approach to Modelling Human Supervisory Control of Dynamic Systems," *Advances in Man-Machine Systems Research*, Vol. 1, William B. Rouse, Editor, JAI Press,Inc., Greenwich, Conn., 1984.

21. Sheldon Baron, Carl Feehrer, Ramal Muralidharan, Richard Pew and Paul Horwitz, "An Approach to Modeling Supervisory Control of a Nuclear Power Plant" NUREG/CR-2988, ORNL/SUB/81-70523/1, Oak Ridge National Laboratory, Oak Ridge Tennessee, 1982.

22. Albert L. Stevens, Bruce R. Roberts, et al., "Steamer: Advanced Computer Aided Instruction in Propulsion Engineering," BBN Report 4702, Cambridge, Massachusetts, 1981.

23. See page 229.

24. See the Advanced Command and Control section below for a brief discussion of ALBM.

25. Previous claims for future successes of AI that did not stand up induced many experienced researchers in the community to be cautious about near-term prospects for application.

26. Walter Reitman, Ralph Weischedel, Kenneth Boff, Mark E. Jones, and Joseph Martino, Automated Information Management Technology (AIM-TECH) Considerations for a Technology Investment Strategy," AFAMRL-TR-85-042, Wright Patterson AFB, Ohio, May 1985.

27. Robert Schudy, Bruce Wilcox and Richard Shu, "Artificial Intelligence Applications Study," AFAWL-TR-85-1140, Wright Patterson AFB, 1985.

28. Robert Schudy, Rusty Bobrow, Ken Anderson and Bruce Wilcox, "Avionics Expert Systems Study," AFWAL-TR-86-1051, Wright Patterson AFB, Ohio, 1986.

29. The name of the department changed a couple of times over time but the overall activities and leadership remained fairly constant.

30. The inclusion of various parallel computers was in part mandated because CASES development was an element of, and funded by, DARPA's Strategic Computing Program. However, the speed-up of execution of some of the more complex models was an important factor in providing timely results.

31. By this time, Mike Dean had moved from Cambridge to the Northwest to accompany his wife who had a professional opportunity there. Mike was a highly valued member of BBN, and was demonstrably capable of working on his own for extended periods, so we suggested he remain with BBN and work out of his home. He was re-assigned to the San Diego office because the work in Command and Control had become a principal focus of his and because the West Coast office location made communication and interaction much easier because of distance and time-zone considerations.

Chapter 10

50 Years of Acoustic Signal Processing for Detection

Edward Starr and Richard Estrada

The authors describe their experiences, over five decades, in the digital signal processing revolution. Collaborating with BBN scientists from other disciplines, they have been challenged to find the best technical solutions to a given problem. The area addressed is acoustic signal processing for detecting the sources of acoustic energy. Examples are monitoring airport noise and detecting sound from submarines in the oceans.

10.1 Coping with the digital revolution

Over the past five decades, scientific and engineering advances in one field have had major impacts on advances in other fields. Notably, the massive increase of computing power and the increase of available data storage have significantly affected many scientific and engineering endeavors. Processing physical signals, such as sound or vibration, for the purpose of understanding and/or detecting the sources, is one of these. For example, where once we hauled large analog instruments and tape recorders to the field to record sound and vibration signals, later spending long hours doing playback analysis, such work can now be performed in real time with a laptop computer. Digital processing's explosion of capabilities over these decades has had a dramatic impact. But the path wasn't always smooth. In this article we will describe, from personal experiences, how the digital revolution transformed the detection and measurement of acoustic signals for the purpose of monitoring or locating and classifying the sources — examples being aircraft and submarines.

Because of the long-standing technical diversity within Bolt Beranek and Newman (BBN), cross-fertilization between technologies frequently occurs, planned in some situations but more often ad hoc. The bridge between digital computing and signal processing for the physical sciences was one of these cross-fertilizations. Each of the author's started their careers in different BBN divisions, one in physical sciences and the other in computer and information sciences. We often discussed technical issues together, and later in our careers collaborated on projects described in this article.

As an example of BBN's technical diversity, disciplines that have been applied to some or all of the systems described in this article include noise generation by mechanical systems, psychoacoustics, sound propagation, acoustic arrays, networking and distributed systems, human factors, artificial intelligence, genetic optimization, computer-aided learning, and the architecture of real-time computer systems.

The following sections begin by describing the classic analog approach to acoustic processing that existed in the 1950s and early 1960s and then migrate to examples of hybrid systems using a mixture of analog and digital capabilities. Next, we present experiences in digital signal processing when it was in its infancy and track advances and improvements. Lastly, we discuss examples of experimental and operational digital processing systems where the systems are all digital except for the sensors and CRT displays. BBN's experience in this area is by no means unique. Rather, we were one of a number of organizations whose work evolved along similar paths to those described here.

10.2 1950s and early 1960s

Analog instruments available to measure acoustic and vibration data were designed, manufactured, and offered commercially by companies such as General Radio in Concord, Massachusetts, and Bruel & Kjaer in Denmark. These products included sound level meters, octave band analyzers, and microphones and accelerometers as sensors. Many of these battery-operated instruments were about the size of a breadbox or two, and weighed many pounds. [1] Data were read from meters and manually written down by the project investigator. One-of-a-kind laboratory analog instruments were also designed to do more sophisticated analyses. Examples are tunable very-narrow-band filters for harmonic analysis and large rotating magnetic drums with variable delay heads to provide time delays for correlation analysis. A small group of BBNers led by George Kamperman that included Denis Noiseux, Herbert Fox, and Ed Starr specialized in the design of such instruments. [2,3]

Large multi-channel analog magnetic tape recorders, filling a full rack, were the primary data recorders for large sets of physical data. Smaller, portable, single channel reel-to-reel tape recorders were used for an individual channel of acoustic or vibration data.

Underwater sensor systems, such as the Sound Surveillance System (SOSUS) [4] developed by Bell Laboratories, provided analog acoustic monitoring capabilities in the oceans for ship surveillance. Hydrophones mounted on the ocean floor sent signals via underwater cable to shore processing stations. Here the data were spectrally analyzed by analog instruments and recorded on paper strip charts as sonargrams (see the sample in Figure 10.7).

In the decades of transition from analog to hybrid to (almost) all-digital systems, BBN, as well as other organizations, applied many strategies to make the best use of the then current state of the art of the two technologies. Below we give two examples of hybrid analog/digital systems developed during this period.

10.3 1960s and early 1970s

BBN purchased its first digital computer, an LGP-30 [5], in the early 1960s. Shortly after, BBN acquired the initial PDP-1 [6] developed by Ken Olsen and the Digital Equipment Company (DEC). Use of digital computers, even with their limited real-time capabilities, followed quickly. To circumvent the real-time limitations, analog front ends were used to analyze and integrate the analog data, reducing the information to a slowly varying signal that could then be handled in real time by the fledgling analog-to-digital converters. The computer was then used to format and sort the data as needed, print it, and display the information to human observers. Two projects that used this approach in the late 1960s and early 1970s were airport noise monitoring systems and a measurement system to support a joint U.S. and U.K. sonar research project on the HMS Matapan.

Airport noise monitoring systems

In the late 1960s, jet noise from commercial airline operations became a significant annoyance to residents living around major airports, creating a volatile political issue for airports. Jet engines generate more noise energy at higher frequencies than turbo or reciprocating engines, and the energy at these high frequencies contributes more to the annoyance than low frequency energy at the same level. Working with the Port of New York Authority (PNYA), BBN researchers explored human reaction to jet noise. BBN psychoacousticians — Karl Kryter and Karl Pearsons — conducted laboratory experiments with bands of noise to analyze a human's response, and experimentally determined curves of equal human annoyance versus frequency and noise level. [7, 8] These weighting curves were used along with a loudness summation procedure to produce Perceived Noise Level (PNL) in units of perceived noise decibels (PNdB). The procedure was then validated using recordings of aircraft noise in subjective tests. A single weighting curve chosen from the results at high noise levels has become the D-weighting standard (see Figure 10.1) [9] for sound level meters to provide an estimate of relative PNL.

FREQUENCY RESPONSE weightings of sound-level meters. A—40-phon equal loudness. B—70-phon equal loudness. C—100-phon equal loudness. D—Proposed equal noisiness.

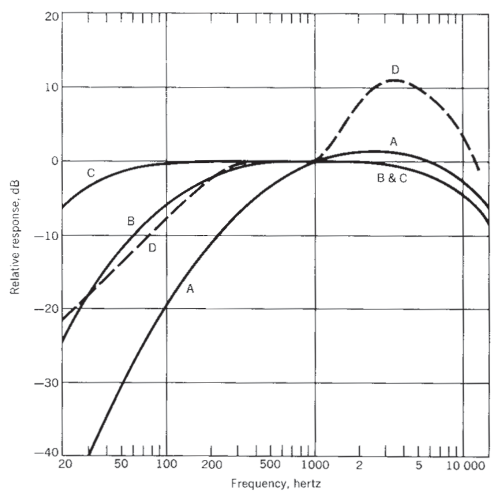


Figure 10.1. D-weighting curve for the measurement of perceived noise. A, B, and C are equal loudness curves. [10]

After considerable political action by residents living near the airports, PNYA decided to install continuous (7 days per week, 24 hours per day) noise monitoring systems using the perceived noise weighting at three New York airports: Kennedy, LaGuardia, and Newark. BBN designed, implemented, installed, and maintained these systems.

Hydrophones (waterproof microphones) were located on telephone poles at strategic locations below flight paths. The remote electronics for the systems were installed in tamper-proof cabinets high on the same pole. In the remote electronics, the instantaneous sound signal was passed through a filter (similar to D-weighting), rectified, integrated, and log-converted. The resulting, slowly varying signal was applied to a voltage-controlled oscillator (VCO). The output frequency of the VCO represented an estimate of the PNL, and the VCO frequency was transmitted over ordinary phone lines to the central station.

At the central station, the frequency from each of the many remote stations was counted and sent to a digital computer (a DEC PDP-8 with 4 Kbytes of memory [11]), which converted the information to a digital PNL level for each of the hydrophone locations each second. (See Bell et al. for a description of the DEC series of computers.[12]) The PNL values were then displayed and printed. Because PNYA used this information to identify individual flights making exceptional noise, the Authority could work with the airlines to reduce levels, and provide an objective rather than emotional basis for addressing the political issues.

In this project, psychoacousticians and electrical engineers collaborated to provide a hybrid analog/digital system that aided the understanding and mitigation of serious noise annoyance problems around the PNYA airports, and later many other airports. After these installations, similar systems were installed at other airports in the United States by BBN and others. BBN staff working on the airport monitoring systems included Byron Blanchard, Joseph Coloruotolo, Robert Coughlan, David Johnson, John Melaragni, Robert Pyle, Edward Starr, and Douglas Steele. [Editor's note: regarding other noise-monitoring projects, see Chapter 8.]

Matapan project

In the early 1970s, BBN worked with the U.K. Ministry of Defense on a major sonar research project for surface ships. James Barger, director of the Physical Sciences Division, led this sonar research project for BBN. To support the sonar research, a shipboard measurement system was needed. A large number of accelerometers and hydrophones were distributed about the test platform, the HMS Matapan (Figure 10.2), to measure the self-noise of the sonar system. Self-noise is the background noise in the sonar system induced by ship vibration, ambient sea noise, and flow noise near the sonar system's receiving hydrophones. A fundamental limitation on the detection level of the received sonar signals, self-noise is greatly affected by the sea state, wind direction, and the ship speed.

In the Matapan Instrumentation System, the signals from hydrophones and accelerometers placed at a great many locations around the ship were cabled to the Scientist's Lab. Individual sensor signals were selected via a patch-panel and a matrix switch. These signals were then sent to a standard analog 1/3-octave-band filter bank. The rectified and averaged output of the filters was digitized and sent to a DEC PDP-11/20 [13] with a large (for the time) 128-Kbyte onboard magnetic disk drive to store the results. The computer organized, calibrated, and stored the data by channel for later viewing. Selected channels could then be displayed on a CRT and as 1/3-octave outputs on an X-Y recorder. The individual data sets were automatically labeled with ship speed and direction, and wind speed and direction, both of which were digitized from ships' sensors. [14].

A block diagram of the system is shown in Figure 10.3. The analog instrumentation is on the left side prior to the PDP-11/20. The 11/20 was running the RT-11 operating system. Figure 10.4 shows the twelve racks of equipment that were required to perform these tasks in the early 1970s. Individual equipment is identified in the caption. DEC-tape (small magnetic tape) was the medium for loading software as illustrated by Bob Pyle in Figure 10.5. Much of the individual analog equipment (vintage early seventies) is shown in Figure 10.6.

Figure 10.2. Photograph of the HMS Matapan, platform for the Joint U.S. and U.K. Sonar Research Project. The Matapan Instrumentation System is located in the Scientist's Lab on the forward main deck. (Photo courtesy of Douglas Steele and the Royal Navy Admiralty Underwater Weapons Establishment.)

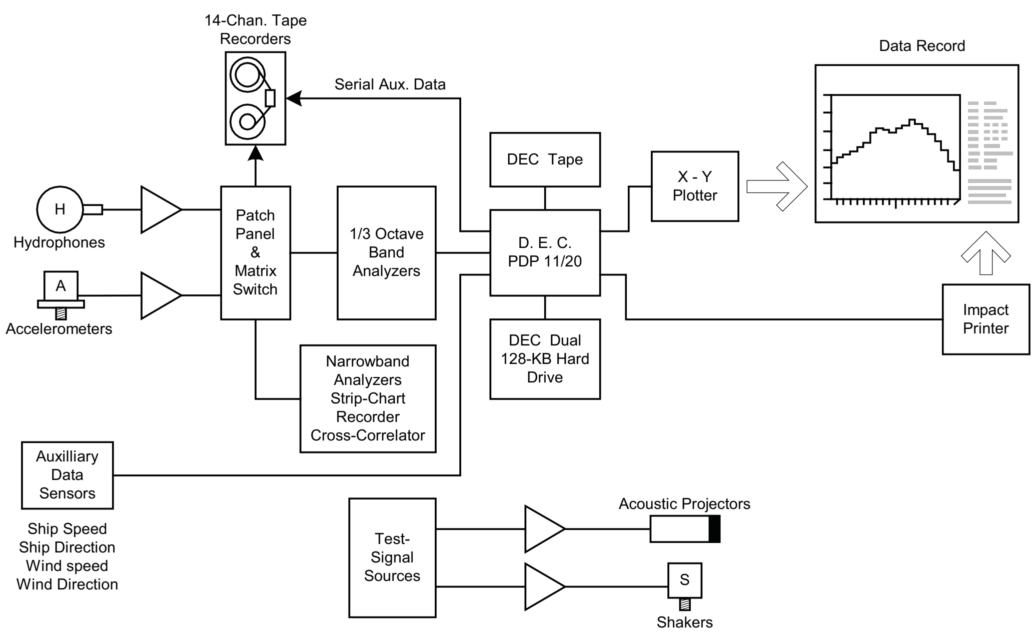


Figure 10.3. Block diagram of the Matapan Hydro-Acoustic Instrumentation System, an example of an early hybrid analog/digital system. (Diagram courtesy of Douglas Steele.)

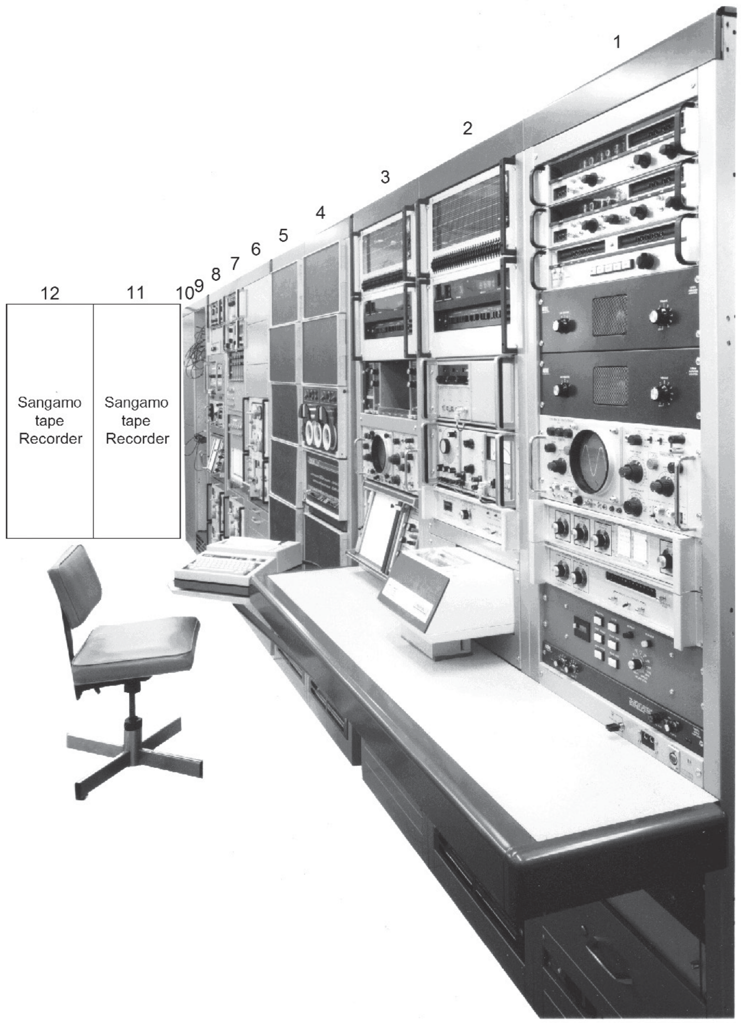


Figure 10.4. The hardware for the Matapan Instrumentation System required a dozen racks. Rack 1: Systron Donner clock/timecode generator, Altec Lansing monitor speakers, Tektronix RM-503 oscilloscope, reed-relay matrix switch control, and Sangamo tape recorder control; Rack 2: General Radio 1921 1/3-octave band analyzer, Hewlett Packard test equipment, and impact printer to annotate XY chart data; Rack 3: Another General Radio 1921, Ithaco analog amplifier, Tektronix RM-503 oscilloscope, XY chart recorder; Rack 4: synchro-digital converter, analog-digital converter, dual DECtape drives, DEC PDP-11/20; Rack 5: two 64-MB hard disk drives; Racks 6-8: See Figure 10.6; Rack 9: MAC plugboard analog patch panels; Rack 10: hydrophone and accelerometer amplifiers and reed-relay matrix switch; Racks 11-12: Sangamo 14-channel instrumentation tape recorders. (Photo courtesy of Douglas Steele and BBN Technologies.)

Figure 10.5. Bob Pyle loading magnetic DECtape for the PDP-11/20. (Photo courtesy of Douglas Steele.)

BBN installed the system onboard the HMS Matapan in Portsmouth, UK, and operated the system during trials. In the first voyage with the system installed, the Matapan sailed from Portsmouth around Land's End and through the Irish Sea in Sea State 6 (very rough seas) to Loch Fyne in Scotland. Here, baseline stationary sonar self-noise measurements were made, and the overall system was checked out. Measurements over a wide variety of conditions were conducted in support of the primary sonar research mission over the next three years.

The Matapan Instrument System was an example of the supportive collaboration among professionals in acoustics, mechanical and hydrodynamic noise-generating mechanisms, sea-trial design, and computer programming and system design. This was also an example of a hybrid system that made use of the best available analog and digital capabilities at the time to accomplish a specific mission. This project also provided BBN researchers with a lot of sea time, which, although requiring long work hours, was exciting work.

Many participated in the experimental phases of Matapan, including Jim Barger, Robert Wagner (deceased), John Marchment of the Royal Navy Admiralty Underwater Weapons Establishment, Robert Vachon and John Hammond of the U.S. Naval Ocean Systems Center (NOSC), Robert Collier, John Lorusso, John Melaragni, Robert Pyle, Edward Starr, and Douglas Steele.

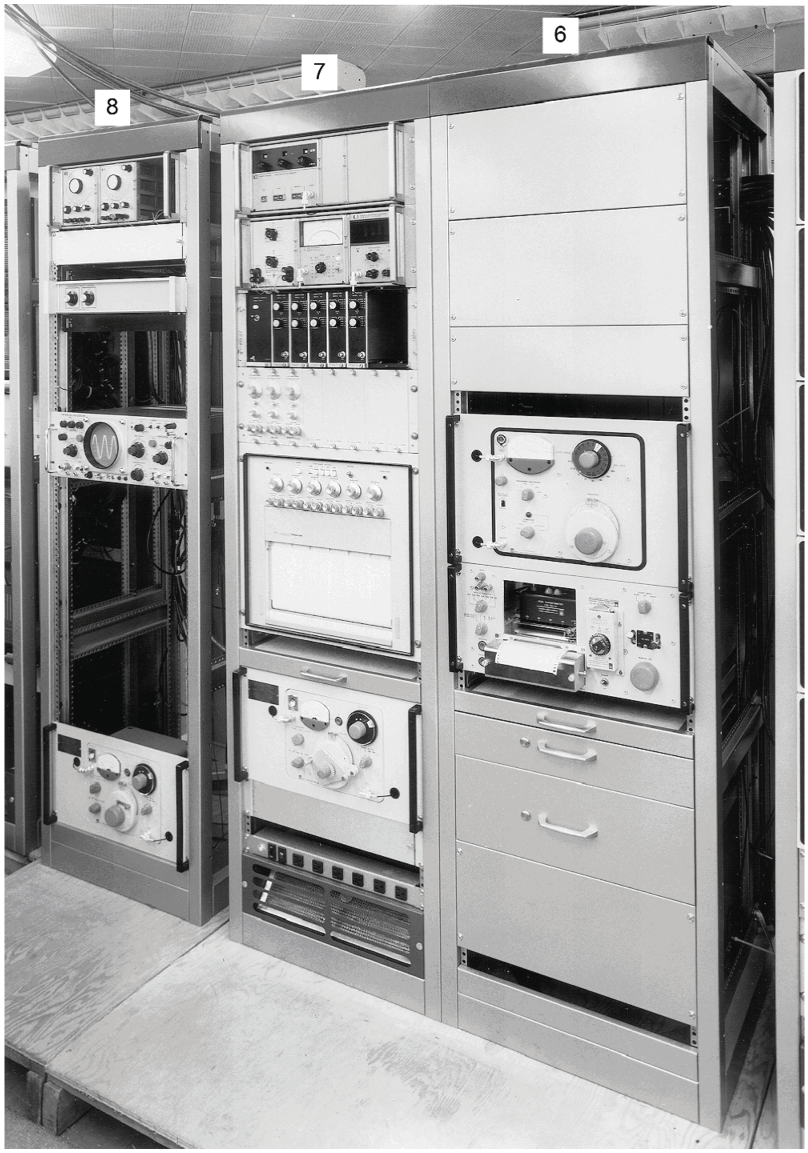


Figure 10.6. Photo of the commercial analog test equipment mounted in racks 6 to 8. Rack 6: General Radio narrowband analyzer and graphic level recorder; Rack 7: Hewlett Packard (H.P.) test equipment, Ithaco analog amplifiers, Gould/Brush 280 strip chart recorder, General Radio tunable 1/3-octave band analyzer; Rack 8: H.P. test equipment, spare matrix switch control, (not shown: time-domain correlator), Tektronix RM-503 oscilloscope, and spare tunable 1/3 octave band analyzer. (Photo courtesy of Douglas Steele and BBN Technologies.)

10.4 Undersea surveillance signal and information processing basics

The task of passive acoustic surveillance can best be understood with a simple analogy. We are all familiar with the "cocktail party" problem in which there are many people in a room having multiple conversations. We hear snippets of many different conversations. We could imagine placing microphones around the room and then trying to focus on the several key conversations we wanted to hear. In the ocean ships don't talk, they hum. The humming comes from their rotating machinery. In acoustic surveillance we placed hydrophones around the ocean to listen to the hums and try to find out who was humming with a Soviet accent as represented by combinations of hums.

As signals, the sounds from the rotating machinery, much of which is speed dependent, yield harmonic lines (hums). The traditional method of displaying this information to an operator is intensity as a function of frequency versus time, often called a sonargram (see Figure 10.7). On such a display, a rotating machine running at constant speed would appear as a straight horizontal line (a hum). In a busy ocean, locating the faint line of a potential submerged target among all those generated by surface ships and fishing vessels is a very challenging job. An advantage that BBN brought to this puzzle was an understanding of the sound-generating mechanisms, their propagation in the ocean, and the signal and information processing techniques needed for detection.

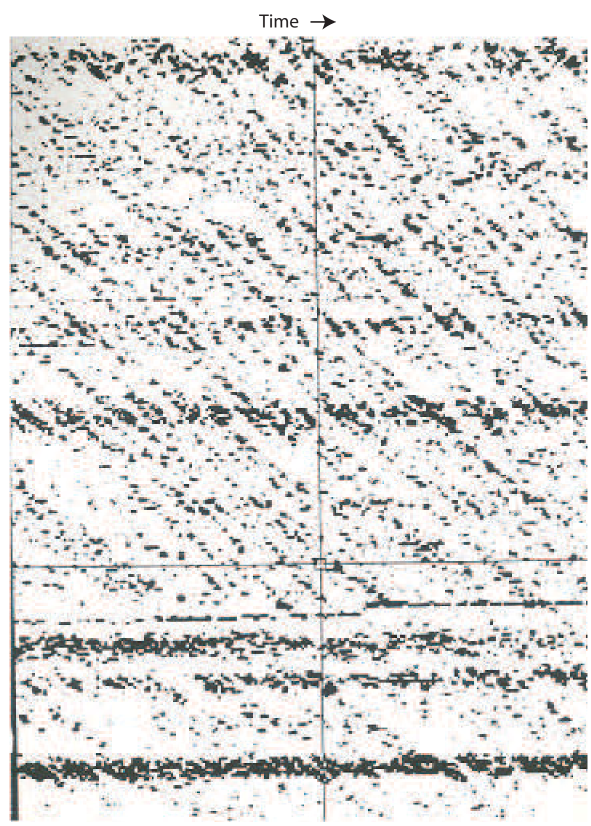


Figure 10.7. Example of a sonargram: time (horizontal axis) vs. frequency (vertical axis). (Photo courtesy of the authors.)

Underwater microphones (hydrophones) produce continuous analog signals that can be sampled to produce a single time-series for each hydrophone. The hydrophones can be mounted on ships, towed behind ships (towed-line arrays), placed on the bottom of the ocean, or be part of a floating sonobuoy. Typically, a number of hydrophones will be placed in a geometric configuration: a hydrophone array. The spacing and topology of the hydrophones will be determined by the frequency of signals being processed as well as by mechanical engineering requirements to maintain the array shape. Like any antenna, the more hydrophones placed in the array, the stronger the ability of the system to pick up distant or weak sounds.

The general form of acoustic data processing has changed little over the past 50 years, although the specific algorithms have evolved considerably. Figure 10.8 shows the typical stages of acoustic processing of the type described in this article. The first stage

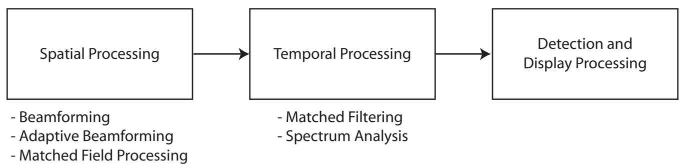


Figure 10.8 Typical stages of acoustic processing. (Photo courtesy of the authors.)

is spatial processing: the hydrophones' time-series are combined to emphasize signals coming from a certain area or direction. The simplest version of spatial processing is beamforming. Here the signal model is a plane wave approaching the array from a specific direction. The hydrophone signals are delayed and summed to form a beam time-series. The delays are chosen to be the delays of a plane wave arrival if coming from the selected direction. Many beams can be formed simultaneously to look in many directions at once. There can be much more complex spatial processing than simple beamforming. For example, one can model the noise field in the vicinity of the array and optimize the signal-to-noise ratio for signals coming from a specific area or direction.

The second stage of the signal processing is temporal processing. This is usually matched filtering in the case of an active system or some form of spectral processing in the case of a passive system. In an active system, noise emitters send out signals of varying shapes and character. The temporal processing compares the incoming data to the transmitted signal, compensated for its travel time and the likely echo characteristics of the target. A passive acoustic system listens for the sounds emitted by the targets; we use spectrum analysis to select tonal sounds or signals in frequency bands. The signal processing results can be displayed directly to operators and/or undergo additional computer information processing to make detections and then classify and track targets automatically. Between signal processing and information processing there is the potential for many different types of algorithms and procedures. It is these candidate algorithms that were implemented and tested in the research activities we describe in the following sections.

10.5 Early all-digital signal processing

In the early 1970s, the processing power of digital computers was not adequate to perform real-time digital processing for projects such as the Matapan system. Even though, starting in the late 1960s, acoustic processing research was being conducted with completely digital signal processing. At that time and into the early 1970s, spectrum analysis became practical in software with the invention of the fast Fourier transform (FFT) [15] and its implementation as a Fortran-callable subroutine. This facilitated spectral analysis of digital-time series that had been digitized from analog tape recordings.

The development of new ways to detect and track targets in the water with sound required the collaboration of many disciplines; physical and engineering analysis of the sources of the signals (sometimes based on limited information), sound propagation solutions, and synthesis of new algorithms based on applied signal processing theory. These algorithms were implemented in software and tested on real acoustic data collected from hydrophones. The algorithms were then incrementally modified and improved based on the experimental results. Only when a new algorithm was thoroughly proven with real-world data would it be considered for installation in an operational system such as SOSUS. [4]

To conduct research for processing acoustic data in new and novel ways, we had to record the data on analog recorders, and digitize and analyze the data on general-purpose digital computers. We could slow down the playback of the tape recorders to a speed compatible with the available A-to-D converters. But, of course, slowing the data rate then required much more time to get an adequate data set. Our ability to create and test new algorithms on acoustic data was severely hampered by two shortfalls. The first was the slow and tedious process of batch processing with punched-card inputs. Often, when a new algorithm or subroutine was to be implemented, keypunch and compilation errors would take several days to find and fix. Runtime errors were addressed by inserting many print statements, which wrote out interim results as the run progressed. When the program crashed, an octal or hexadecimal dump of memory was printed on a long ream of line printer paper. Analysis of these dumps required unusual (some might even say twisted) mental processes. Analysis of such dumps was sometimes referred to as "looking among the ashes of a fire to see what had been cooking." Since only one or perhaps two runs per day could be made with batch processing, it could take many weeks to debug a new processing idea.

The second shortfall was the limited amounts of acoustic data that could be processed with the available computer power. Running spatial processing and spectrum analysis on an early 1970s general-purpose computer (such as a UNIVAC 1108) [16] was extremely time-consuming and it might require months of work to process a few hundred hours of data from a single sensor array. Although lots of data could be recorded on multi-channel analog tape recorders, months could pass before we could process even a small percentage of the data through a train of processing algorithms. Also, since researchers often picked the best pieces of data to process, people in the operational community were concerned that new algorithms that performed well on a few selected pieces of data might not remain valid after being built into an expensive, hard-wired real time operational system.

During the early and mid-1970s, BBN staff used the DEC PDP-10 [17] running TENEX [18] software to develop and evaluate new acoustic processing algorithms to alleviate many of the problems with batch processing. The Text Editor and Corrector (TECO) text editor, absolutely awful by today's standards, was hugely better than using punched cards. Time-sharing allowed the developer to compile and build a program during the time it took to get a cup of coffee. Most important, TENEX had a symbolic debugger, which allowed the developer to set breakpoints, examine arguments, and even patch the code if a problem were found. Thus, an idea could be turned into implemented software within a day or two. However, even with these major improvements, it still could take weeks to process sufficient amounts of acoustic data.

Acoustic Research Center: 1975-1985

The mission of the Defense Advanced Research Project Agency (DARPA) was and is to push the technical frontiers of science to solve problems important to the Department of Defense (DoD). During this period, Soviet ballistic missile submarines were a major threat to the United States, and the capability to track them and know where they were was a national priority. In 1975, DARPA established the Acoustic Research Center (ARC) at Moffett Field in Sunnyvale, California, to advance research in ocean acoustic signal and information processing.

BBN, along with other companies, had a series of DARPA research contracts to work at the ARC over the 10-year life of the center. The ARC could remotely access real-time ocean acoustic data and bring that data over satellite links into the facility. The goal was to create prototype signal and information processing algorithms for use in real-time experiments and in rapid post-analysis. These proven functions could then be transitioned into operational submarine surveillance systems.

The initial computer suite at the ARC was beyond the forefront of technology, leaving much to be desired both in reliability and performance. Most of the computers were either one-of-a-kind or had been substantially modified to meet the ARC's requirements. Most computer scientists who were working elsewhere during the 1970s wouldn't recognize any of these computers with the exception of the PDP-10.

The basic ARC architecture was divided into two parts, as Figure 10.9 shows. The Data Gathering Subsystem selected data streams of interest from each remote site, processed them to select frequency bands and channels of interest, and then transferred the data in real-time over satellite links to the ARC. The Analysis Subsystem at Moffett Field's central site performed real-time and near real-time signal processing, information processing, [19] and displayed the results.

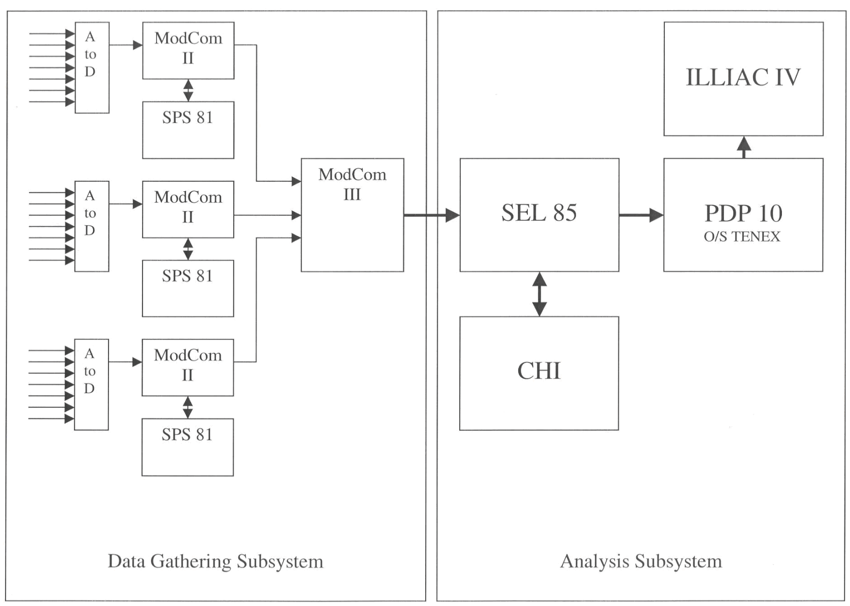


Figure 10.9. Diagram of the Acoustic Research Center, 1975 to 1978. (Photo courtesy of the authors.)

The core of the data gathering subsystem was located at the remote sites. These computers had access to the analog hydrophone and beam signals available at the site. The Modular Computer Corp.'s [20] ModCom II minicomputer's job was to select the channels of data to be processed, digitize them, and use the local Signal Processing Systems' SPS-81 [21] signal-processing computer to perform beamforming and frequency band selection. The ModCom II minicomputers were difficult to use because they were programmed in assembly language. The SPS-81 required the user to write in microcode, again very difficult. In addition, the SPS-81s were plagued by hardware problems. After a year or two, the SPS-81 computers were bypassed completely and the raw time-series data was sent directly to the ARC facility via satellite. The ModCom III (a larger version of the ModCom II) was the final piece of the data gathering subsystem; it buffered the data in memory before sending it to the analysis subsystem.

In the analysis subsystem, the Systems Electronic Lab's SEL 85 was planned as the real-time data manager, recording and buffering data on disk and controlling the Chi signal-processing computer. Fortran programs on the SEL could make subroutine calls to transfer data to the Chi. Other calls would instruct the Chi to perform a sequence of signal processing operations drawn from a large software library. The parameters of the signal processing functions could be set at runtime, so the functionality could be changed from one run to the next.

Unfortunately, the SEL-85 performed poorly at the data manager job, and it was often down. The ARC's SEL-85 had been modified in an undocumented way, making it unreliable and difficult to maintain. The Chi, an early-generation signal processor, had a brilliant design for the time, but had hardware reliability problems. The PDP-10 running TENEX as an operating system was initially the only reliable computer at the ARC, but it had limited processing power. However, as discussed, it provided a good environment for building and debugging signal and information processing programs.

In 1976 there was of course no Ethernet, and generally every connection between two computers was a one-of-a-kind affair. At the ARC, the connections between the SEL and the PDP-10, and between the SEL and the ModCom III, were technological adventures, which often had problems.

The first experiment at the ARC was in summer 1976. During this experiment, which lasted a month, only a few hours of useful data were recorded and analyzed, not unusual for an initial operation of advanced systems. Interestingly, numerous press reports in the late 1970s said that massive amounts of processing power had been assembled at the ARC and that it was making the oceans "transparent," implying that any submarine in the ocean could be found. For researchers at the ARC, living with the reliability issues of these machines, these stories were always extremely humorous. It was said, "There must be another ARC at some other place doing this work" — precious little data analysis was being conducted at Moffett Field's ARC.

Most of the results in signal and information processing research at the ARC during this early period came solely from the PDP-10. Even if it took a whole weekend to process a limited amount of data on this modest computer, it was much easier than getting the SEL/Chi/PDP-10 configuration to work for several hours. Having all the computers shown in Figure 10.9 working at the same time seemed miraculous. [22]

In 1979, the SEL and Chi were replaced by the next-generation machines, which were more reliable. These were a DEC PDP-11/70 and Floating Point Systems (FPS) AP-120B. [23] This greatly improved operations at the ARC, and now we could perform reasonable real-time experiments. But the PDP-11/70, running the real-time operating system RSX-11M [24], had a very small address space; any reasonable program required in-memory overlays, which were difficult to construct and debug. However, the computer was reliable and was good at real-time multitasking.

The AP-120B signal-processing box was a real computer. It was reliable, there were many of them manufactured, and it had an extensive software library of signal processing functions that could be called from Fortran programs in the PDP-11. The simplest way to use the AP-120B was to write a program in the PDP-11, which made a call to signal processing functions, one after another, with indexing and looping implemented in Fortran on the PDP-11. However, when used this way, the AP-120B was idle most of the time while control moved back and forth to the PDP-11.

FPS had a primitive Fortran-like language for building larger signal processing operations in the AP-120B, with looping and indexing inside. These larger modules could be built on the PDP-11 and downloaded to the AP-120B at runtime. However, the AP-120B compiler was unforgiving, with limited debugging capabilities when executing. Nevertheless, now large, fast, and flexible signal processing functions could be run in real time, and engineers could accomplish post-analysis of a significant amount of data.

Eventually, the PDP-10 and 11/70 combination was replaced by a DEC VAX/VMS system. [25, 26] The VAX had a better development environment than the PDP-10 and was faster and better at real-time operation than the PDP-11. With a VAX/AP-120B combination, the ARC was able to conduct a significant number of valuable real-time experiments.

The ARC work was bypassed by technology and the facility was closed around 1985. But the efforts conducted at the ARC stimulated a whole series of signal processing techniques that were used in operational systems later, 5 to 10 years in the future, and indeed generally improved the signal processing capabilities and techniques used in operational acoustics surveillance systems. Further, the ARC work showed the value of building flexible signal processing architectures that could easily be configured from one experiment to the next, and it showed the importance of reliability in complex systems. BBN's architectural software experience at the ARC was of much use for work done in the mid-1980s and early 1990s.

Over the years, many BBNers contributed to the work at the ARC, including Jeff Berliner, Doug Cochran, Charles Epstein, Dick Estrada, Tom Fortmann, Tom Hammel, Steve Milligan, Ron Mucci, Lynne Omelcheko, Ron Scott, Jeanne Secunda, Mike Sullivan, and Jim Sulzen.

AUSEX Project: 1977-1979

In the mid-1970s, analysis and laboratory experiments conducted by Jude Nitsche and John Waters [27] and later by James Barger, Jude Nitsche, and Dave Saches [28,29] at BBN suggested that low-frequency, narrow-band propeller noise from turboprop aircraft and helicopters could be transmitted efficiently through the air/water interface and propagated into the ocean's depths. This led to the hypothesis that a submerged submarine with the right apparatus should be able to detect an antisubmarine warfare (ASW) aircraft (usually turboprop powered) overhead at significant ranges. AUSEX (which stood for Acoustic Underwater Sound Experiment) sponsored by DARPA, was conducted in the late 1970s to test this hypothesis with a full-scale experiment conducted at sea with ASW aircraft. Wesley Jordon of the U.S. Navy was the initial project officer for DARPA, succeeded by Bob Bartlett. The AUSEX project was a collaboration of research in sound generation and propagation by physicists and a feasibility experiment implemented by computer systems, human factors engineers, and by experts in sea trial design.

BBN conducted a multi-week demonstration at the Barking Sands Test Range near Kauai, Hawaii, in 1979. The RV Washington (see Figure 10.10), a Scripps oceanographic vessel, was outfitted by BBN with a submarine towed-line hydrophone array to be the surrogate submarine. The system was staged at the University of Hawaii pier in Honolulu (see Figure 10.11), and loaded onboard the RV Washington (see Figure 10.12).

The AUSEX demonstration system was designed, built, and installed by BBN. This project used hardware similar to that being installed at the ARC at the time (PDP-11s and FPS AP-120Bs). However, as a focused, single-company effort, the AUSEX project

Figure 10.10. Scripps Institute of Oceanography Research Vessel Thomas Washington, which served as a surrogate submarine while towing a submarine acoustic array. (Photo courtesy of Joe Walters Jr.)

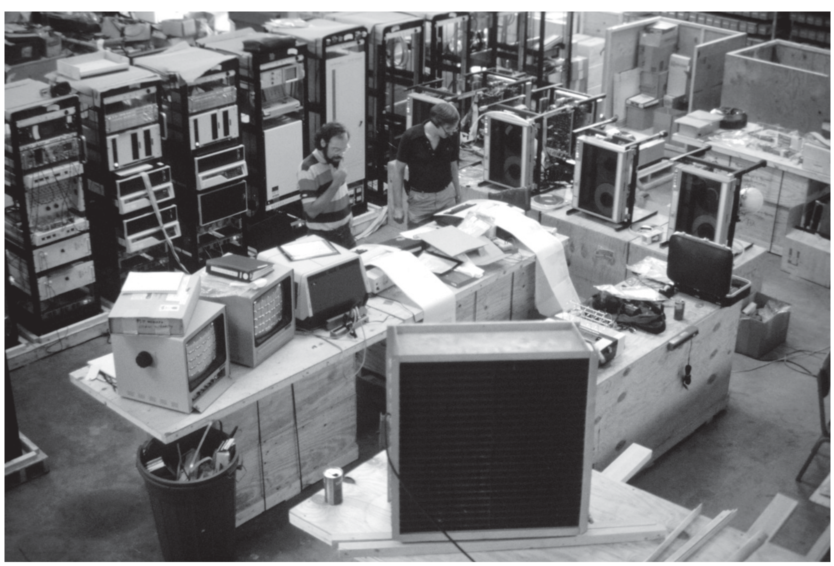


Figure 10.11. BBNers Doug Steele and Steve Blumenthal check out the AUSEX system after transit from Cambridge at a University of Hawaii warehouse dockside before installing onboard the RV Washington. (Photo courtesy of Joe Walters Jr.)

Figure 10.12. Submarine towed-line array installed in its winch is lifted aboard the RV Washington. (Photo courtesy of Joe Walters Jr.)

was able to build and install the signal processing and display system much faster (20 months from start to demonstration) and with more modest funding.

Figure 10.13 shows a block diagram of the system. [30] Analog data functions for the towed array (power, instrumentation, and signal recovery) were handled by a modified BQQ-5 receiver. [31] The large number of analog hydrophone channels from the towed array were digitized, saved on a 9-track digital tape, and fed to an FPS AP-120B signal processor for narrow-band analysis and beamforming. These processes were managed by a PDP-11/34. A second PDP-11/34 formatted and displayed the data to operators, whose job was to detect the narrow-band signature of ASW aircraft from the displays. Figure 10.14 shows a photo of some of the equipment during installation on board ship.

Multiple-hour test sequences were conducted at random times of an extended day (5:00 a.m. to 10:00 p.m.) over a period of two weeks. The detection crew was located in a closed compartment onboard without visual or audio access to the exterior world (see Figure 10.15). There were many periods without aircraft to collect false alarm information. Missions were flown by United States Navy P-3 aircraft and helicopters at unpredictable times. Many successful detections were achieved at tactically significant ranges and the false alarm rate was reasonable, validating the original hypothesis. [32]

BBN members of this system development and experiment team included Hawley Rising and Paul McElroy (both deceased), John Bigler, Steve Blumenthal, Howard Briscoe, Kathy Jones, John Knight, Charles Malme, John Melaragni, Jude Nitsche, Edward Starr, Douglas Steele, Kenneth Theriault, and Joseph Walters.

10.6 Mid- and late 1980s

BBN undertook a number of shipboard signal and information processing experiments in the mid-to-late 1980s. These experiments required much more processing power than was available in a single FPS AP-120B array processor. By this time, the FPS AP-120B was replaced by the FPS Model MP32. Each MP32 had multiple arithmetic units,

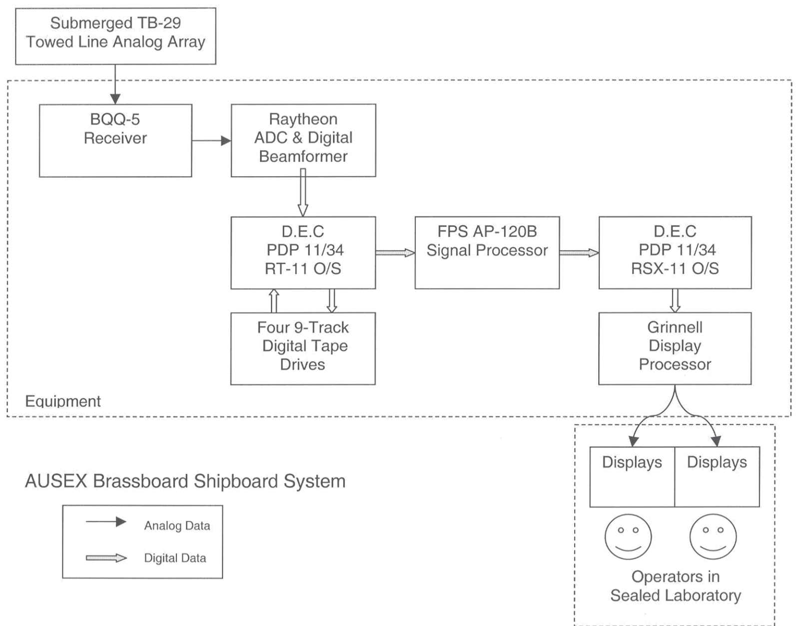


Figure 10.13 Diagram of AUSEX System. (Photo courtesy of Joe Walters Jr.)

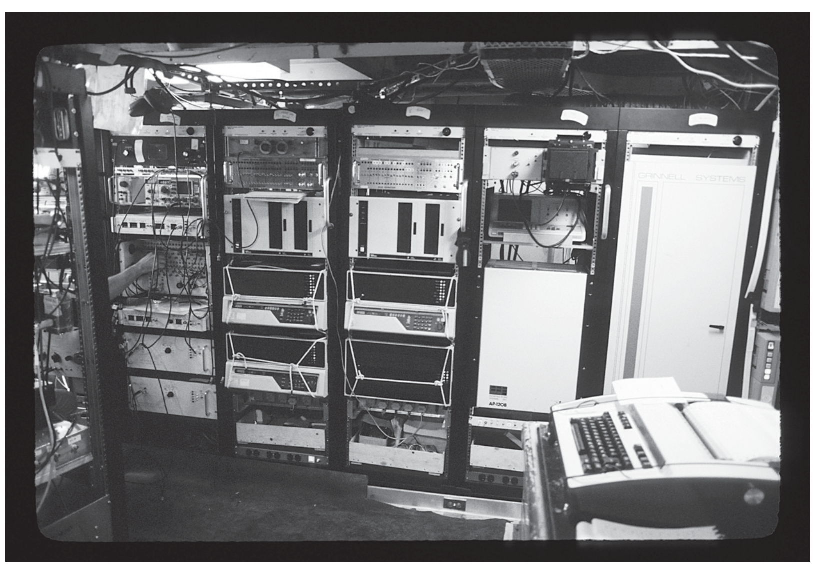


Figure 10.14. Photo of some components of the AUSEX system installed aboard the RV Washington. In the first four racks from right to left; Grinnel Display Processor, AP-120B, and the pair of PDP-11/34s. (Photo courtesy of Joe Walters Jr.)

Figure 10.15. A detection watch, located in a sealed cabin, is on duty. This BBN watch team consisted of Howie Briscoe, Kathy Jones, and Ken Theriault. Note the dartboard for the occasional resolution of ambiguities. (Photo courtesy of Joe Walters Jr.)

compared with the single unit in the AP-120B. However, an MP32 still had a memory limitation. The processing required for a large array of sensors needed at least the power of three MP32s.

While VAX processors during this period were somewhat powerful, they could not possibly keep multiple MP32s busy when interim processing results needed to go back and forth between the VAX and the MP32s. In 1985 Aptec, a small company, brought out a product that provided a large, fast memory for buffering interim signal processing results and handling fast transfers between itself and the MP32s. Hydrophone data could be digitally sampled and sent directly to the Aptec memory. (see Figure 10.16). After some buffering, chunks of data would be transferred to an MP32 for beamforming. The data would return to the Aptec system for further buffering, and be returned to the MP32 for spectrum analysis and matched filtering. In the systems we configured, one VAX controlled this processing, and another VAX received the analyzed data, performed information processing, and displayed the results.

The software environment in this configuration was much improved over the ARC, but still fell far short of modern standards. VAX application programs were now written in the C language, a big advance over the earlier Fortran. The VAX/VMS systems had excellent symbolic debugging for the time. The Aptec software was written in a Fortran-like language on the VAX and downloaded to the Aptec at runtime. The Aptec operating system was primitive; the only way to debug it's software and the MP32s was to pull data segments up to the VAX and look at the stream of numbers. Most of the MP32 software was similar to the software previously built for the AP-120Bs, except now one had to worry explicitly about three arithmetic units doing calculations rather than one.

Very complex software had to be written for the VAX, Aptec, and MP32s to control the movement of data from different buffers in the Aptec to the individual memories of the MP32, and then to instruct the MP32 processors when to process data. This was the most important architectural difficulty for this configuration since the mechanisms for controlling these actions were spread over many independent processing streams.

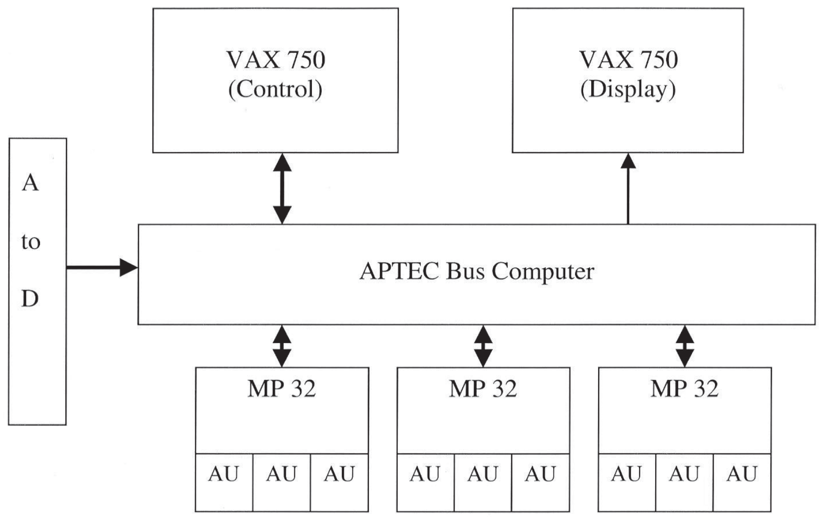


Figure 10.16. Block diagram of a system employing the Aptec Bus. (Photo courtesy of the authors.)

Further, the signal processing functions in the MP32s were written in an unforgiving language.

Overall this system was vastly superior to the one at the ARC, both in hardware and software capability, reliability, and development environment. Indeed, much of the software architecture of the system was based on architectural ideas that had first been used at the ARC and in the AUSEX system. We obtained a number of important experimental results with this system. Members of the BBN team included James Barger, Ed Campbell, Ed Combs, Richard Estrada, Mark Hamilton, Karen Kupres, Ron Mucci, Chris Remer, Sue Riley, Edward Schmidt, Jeanne Secunda, Ken Theriault, and Mary Trvalik.

ARIADNE: 1986-1988

In the mid-1980s, modern Soviet submarines (for example, the Akula class) generated much less noise than earlier Soviet submarines. [33] The U.S. Navy needed a major upgrade to the SOSUS system, which had been in operation for many decades, in order to monitor the quieter boats. This upgrade had two primary goals: the detection of much quieter submarines, and a substantial reduction in operational manpower and therefore costs. This was a challenge of national priority.

The most feasible method for detecting quieter submarines was to place the hydrophones closer to the target. This required significantly more hydrophones and hence a greatly increased processing load. SOSUS was a manual surveillance system: The operators looked at every piece of data on paper and made the detection decisions. Unfortunately, hugely increasing the number of hydrophones would also hugely increase the staffing requirements for a manual system. The ARIADNE project was initiated to find solutions to these problems. The ARIADNE concept was a prototype system using fiber-optic cable to connect a large number of bottom-mounted hydrophones. ARIADNE's goals were to provide technology to detect the quietest submarine and at the same time show the methods whereby staffing could be reduced by a factor of ten over the then current staff.

In 1986, BBN won a contract with the navy to design and demonstrate the shore-processing system for the ARIADNE program. BBN proposed to accomplish the aggressive goals through advanced signal processing, artificial intelligence techniques for information processing, and improved display techniques. Over the next two years, BBN built a prototype ARIADNE system in phases. Upgrades were periodically installed at an operational site where it was used and evaluated by navy operators with real data under operational conditions. [34]

By this time, powerful vector-array processing hardware was available on a VersaModule Eurocard (VME) board [35] such as those produced by Sky Computer [36] and Mercury Computer Systems [37]. The signal processing was all done in a customized VME system named the node cluster processor (NCP), and the hardware architecture was an early adopter of VME-bus technology. Being an early adopter led to some frustrating, early bus timing problems between the boards of the NCP that were very difficult to localize, but after a few months these were solved.

This hardware and software architecture finally successfully addressed the problem of simultaneously executing the signal processing primitives, controlling the signal processing flow, and buffering interim results in memory. In addition, the ARIADNE hardware system was constructed totally of commercial off-the-shelf equipment. The earlier stimulation by the ARC and other activities had brought the development of commercial processing hardware to the point where it could meet the challenging capabilities needed for ARIADNE.

Each NCP consisted of a half-rack VME chassis with five triplets, for a total of 15 boards. A triplet contained a general computing board, a vector-array processing board, and a memory board. Each triplet had a large amount of processing power and memory. All the memory of a triplet was in the same address space, and available to all operations. The general-purpose computers could also reference data being buffered on memory boards in other triplets. The ARIADNE prototype system used two NCPs in its signal processing subsystem.

Steve Milligan of BBN, the technical lead for ARIADNE (and also the subsequent Fixed Distributed System [FDS] program described below), selected this hardware configuration and designed the software architecture for the NCP. Signal processing functions such as narrow-band filtering and beamforming were created in the C language to run in the general processor, but use the array processor for CPU-intensive functions like FFTs. A large library of high-level functions was easily created and debugged.

A second important part of the system software was a data-flow architecture that distributed the real-time signal processing computation to be done for a set of ARIADNE sensors across all five of the NCP triplets. The data-flow architecture allowed virtually any acoustic signal processing flow to be created by merely describing the processing in high-level terms in a file. This file would be "compiled" by the data-flow software at runtime.

The NCP software-hardware combination solved many of the problems that made software development difficult at the ARC and in BBN's prior shipboard experiments. (It is a testament to the value of NCP software concepts and the data-flow architecture that they were still used in some real-time signal processing work in 2004, 17 years after its initial implementation.)

Following the spatial and frequency signal processing, the analyzed data was sent to signal extraction, information processing, and workstation subsystems. (See Figures 10.17 and 10.18). The task — to find the very few spectral lines of interest embedded in the huge field of spectral lines from shipping and fishing operations — was indeed like finding the proverbial needle in a haystack. BBN developed operator tools to improve productivity; alerting to significant events; ability to scroll back through the data to compare with previous events; ability to compare with examples of known signals; ease of creating and distributing reports; and so forth. BBN's work on ARIADNE showed the path to meet the goals of the upgrade to SOSUS.

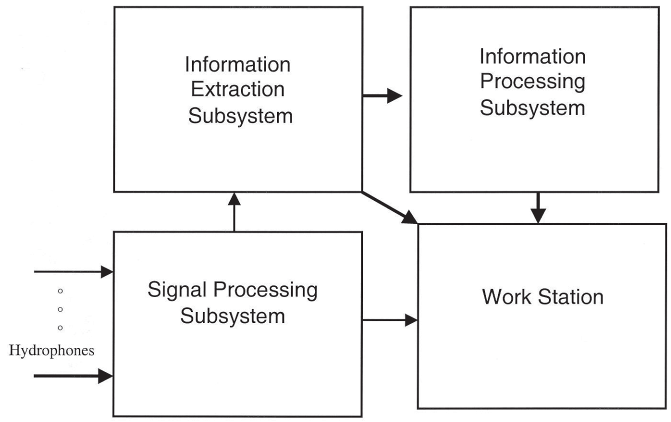


Figure 10.17. Functional diagram of the ARIADNE Shore System. (Photo courtesy of the authors.)

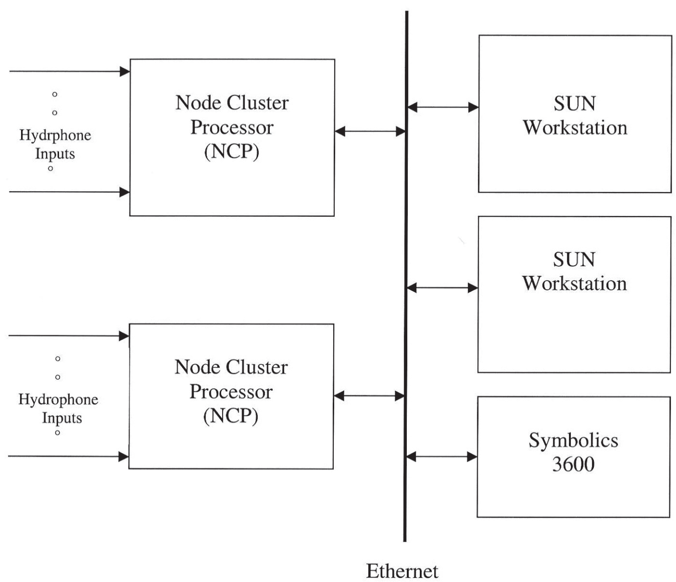


Figure 10.18. Hardware diagram of the ARIADNE Shore System. (Photo courtesy of the authors.)

Key participants from BBN included Mark Berman, John Bigler, Howard Briscoe, Harry Cox, Richard Estrada, Mike Flicker, Kathy Jones, Mike Krugman, Steve Milligan, Jim O'Connor, Ronald Scott, Edward Starr, Douglas Steele, and Jerry Wolf. Roger Harris of NOSC was a key Navy scientific contributor. Ron Mitnick of U.S. Navy Space and Naval Warfare Systems Command (SPAWAR) was the navy program director.

Fixed Distributed System (FDS)

In the ARIADNE project, BBN demonstrated the technology on a small scale to solve the goals for the advanced shore system, reducing the technical risks for the navy. The navy then proceeded to prepare a request for proposal (RFP) for the replacement of the SOSUS system, called the Fixed Distributed System. The FDS Full Scale Engineering Development (FSED) for the shore processing was a competitive procurement for a large system requiring full military specifications that initially included software programmed in ADA, [38] based on then-current Department of Defense policy.

In the early 1980s, BBN had successfully deployed the worldwide operational Defense Data Network (DDN) for the Defense Communications Agency, based on ARPANET technology. However, BBN lacked demonstrated large-system experience with the operational navy. Although BBN had demonstrated the potential technology solutions, could it win a competitive procurement for a major system as the prime contractor? Some inside BBN suggested that the DDN experience was a sufficient demonstration of qualifications for BBN to serve as the prime contractor in the bid for this large system, even though the experience was not directly for the operational navy. Others in BBN disagreed. This led to much internal discussion and debate.

Ultimately, BBN decided to team with a division of IBM Federal Systems (now Lockheed Martin) in Manassas, Virginia, after discussions with other large defense contractors active in this area. The decision was based upon several factors: IBM successfully provided the BQQ-5 sonar systems to the U.S. Navy submarine forces for many years; IBM was looking to broaden its sonar market but had no experience in shore based sonar systems, which gave BBN a stronger position; the strengths of the respective teams were complementary for developing such a system; and the cultures of the two organizations were relatively compatible.

A teaming agreement was signed under which BBN would provide a significant part of the system development and the team would leverage the large-system expertise of IBM Federal Systems Division. The motto for the team for individuals to lead the work was "best of breed." The program manager during the design phase for IBM was Hamilton Walker, and for BBN, Ed Starr. Steven Milligan of BBN was technical lead for the combined team. The program manager for the U.S. Navy was Kirk Evans.

The FDS competition was conducted in two phases. In the first phase, two teams were selected to design the shore system, and these teams competed to win the second phase. The second phase was to design, build, install, and maintain the operational shore systems. Rather than specify a specific navy design, the navy intelligently required the bidders to respond to a performance specification and challenged the bidders to present their designs on how best to meet the performance requirements. In addition, the FDS system was required to be built with commercial, off-the-shelf equipment rather than proprietary hardware.

The Phase One proposal was submitted to the navy in March 1989. There were four teams competing for the design phase, and IBM/BBN and GE-Syracuse each won a contract for the 15-month engineering design competition. The awards were made in October 1989, and the first design review for the IBM/BBN team was on 3 January 1990.

The system designed by the IBM/BBN team was based upon the prototype ARIADNE system. The ARIADNE design was expanded to address the issues of scaling to full deployment, additional information processing and alerting, human factors, training software, and maintenance and logistic issues. In February 1991, the IBM/BBN team was selected by the navy for the multiyear Phase Two full-scale engineering development. For Phase Two, the IBM program manager was Kathy Hegmann, succeeded by Al Simpson. The BBN program manager was Ed Starr, succeeded by Joe Walters.

The NCP architecture was used and expanded. Interconnected workstations were provided for navy operators to handle multiple events, to provide hot spares for reliability and surge conditions, and to support embedded training and maintenance. There were well over 600 computers of various types in the system, performing different tasks ranging from signal processing, display generation, rule-based systems, embedded training, and logistical and maintenance functions. The FDS system design and implementation required the collaboration of many disciplines: computer science, physics, system engineering, logistics, program management, educational technology, and human factors.

The IBM/BBN team successfully completed the development and installation of the FDS system. During the end stages of the development, the international situation changed dramatically with the end of the Cold War. This lowered, but did not eliminate, the national priority for detection of submarines. Key BBN participants in FDS included among many others Mark Berman, John Bigler, Marshall Brinn, Edward Campbell, Michael Flicker, Lacey (Skip) Green, Warren Hollis (deceased), Steve Milligan, David Montana, Rita Reed, Edward Starr, and Joe Walters.

10.7 Summary

In this chapter we have described a rather personal history of the transition, over five decades, from analog to digital signal processing for acoustic signals. This history provides an example of the impact of the huge advances in digital technology on those doing research and development in acoustic signal processing for detection that must parallel the impact in other scientific fields. Scientists and engineers starting their professional careers in the 21st century may be surprised that so much effort was required to do things now possible with a laptop, and the effort to squeeze software and data into storage that is a minuscule fraction of that in an iPod. Less visible, but just as important, are the changes in software architecture, tools, and techniques over the past 40 years.

BBNers made use of these technologies and applied their skills, working with others, to contribute to the solution of technical problems, many of them with national implications, some for the defense of our country and others, such as with airport noise, to improve the quality of life. The motivation for much of the BBN staff was to work on hard problems with others (inside and outside of BBN) who were the best in their fields. Further, the opportunity to interact closely with those working in other disciplines made the work even more exciting. It was hard work, but it made for a rewarding and challenging professional life. All the work discussed here was carried out within BBN's technically challenging (but caring and supportive) environment, and with colleagues who made hard work fun.

Of course, the digital revolution has continued unabated. The technical capabilities in 2008 surpass all expectations 10 or 15 years ago. However, the programmatic and political arenas have not achieved the same progress. Following the demise of the Soviet Union, the investments in ASW have logically declined. But the dissemination of technology continues and has allowed others to achieve a higher plateau, again placing the United States at a potential disadvantage [39]. This is the usual cycle of defense/offence so frequently observed in history. On another plane, annoyance around airports from aircraft has mostly been abated by quieter aircraft engines, airports acquiring properties around the airports, and better management techniques.

The digital revolution continues.

Acknowledgments

The authors gladly acknowledge the numerous contributions of others at BBN and in the government not explicitly mentioned due to faulty memory and "senior" moments. Also, we thank Doug Steele, Joe Walters, Karl Pearsons. Bob Pyle and Jennie Connolly for help in gathering information for this chapter. We also thank the editors, Dave Walden and Ray Nickerson, for the energy and savvy they have contributed to pull this volume together.

References and notes

1. L.L. Beranek, *Noise and Vibration Control*, chapters 3, 4, and 5; McGraw-Hill, 1971, pp. 45-137.

2. D. Noiseux, "Variable Time Delay Drum," BBN Report 1119, Dec. 1963.

3. H. Fox and E. Starr, "Description of BBN 339A Bilateral Log Converter," BBN Report 680, Oct. 1959.

4. For a description and history of SOSUS, see `http://www.fas.org/irp/program/collect/sosus.htmandhttp://www.dosits.org/gallery/tech/pt/sosus1.htm`

5. LGP-30 Librascope General Precision Computer; `http://ed-thelen.org/comp-hist/lgp-30.html#Placard`

6. PDP-1; `http://en.wikipedia.org/wiki/PDP-1`

7. K.D. Kryter, "The Meaning and Measurement of Perceived Noise Level," *Noise Control*, vol. 6, no. 5, 1960, pp. 12-17.

8. K.D. Kryter and K. Pearsons, "Some Effects of Spectral Content, Duration and Rise Time on Perceived Noise," BBN Report 965,Dec. 1962.

9. E. Starr, "Measuring Noise Pollution," *IEEE Spectrum*, vol. 9, no. 6, 1972, pp. 18-25.

10. Work by Fletcher and Munson in the early 1930s had defined equal loudness contours vs. frequency for humans. These curves formed the basis of the other weightings for sound level meters, as shown in Figure 10.1. H. Fletcher and W.A. Munson, "Loudness, Its Definition, Measurement and Calculation," *J. Acoustical Soc. of America*, vol. 5, October 1933, pp. 82-108.

11. PDP-8; `http://en.wikipedia.org/wiki/PDP-8`

12. G. Bell, C. Mudge, and J. McNamara, *Computer Engineering, a DEC View of Hardware System Design*, Digital Press, 1978.

13. PDP-11; `http://en.wikipedia.org/wiki/PDP-11`

14. D. Steele, R. Wagner, and E. Starr, "Design of the Instrumentation Center for Matapan Hydroacoustic Noise Measurements," BBN Report 2162, Mar. 1971.

15. R.B. Blackman and J.W. Tukey, *The Measurement of Power Spectrum from the Point of View of Communications Engineering*, Dover, 1959.

16. UNIVAC 1108; `http://en.wikipedia.org/wiki/UNIVAC_1108`

17. PDP-10; `http://en.wikipedia.org/wiki/PDP-10`

18. TENEX; `http://www.opost.com/dlm/tenex/`

19. Information processing as used in this article generally includes computer-based algorithms for detection of signals, their characterization and classification, grouping of detections which have all come from the same target, and tracking the targets.

20. Modular Computer Corporation: `www.modcomp.com/corparate/history`

21. SPS-81: "The SPS-41 and SPS-81 programmable digital signal processors", Joseph R. Fisher;, pgs 171-174, *Conference record of the 6th annual workshop on Microprogramming*, ACM Press, NY,NY 1973.

22. Some information processing capabilities were intended to run only on the PDP-10. BBN designed one such information processing program to identify and track targets among the

copious signal processing data. The Octopus tracker was the first practical use of Kalman filtering on measurements of uncertain origin (from the target of interest, other targets, or clutter). The Octopus project had actually begun before the ARC program, but the project found a strong home in the ARC. Octopus used a Bayesian approach to associate uncertain, asynchronous, heterogeneous measurements (bearings and time delay differences) with targets, displaying position estimates, and error ellipses on a geographic display. This Probabilistic Data Association Filter became the key algorithm and is described in a textbook co-authored by BBNer Tom Fortmann and consultant Yaakov Bar-Shalom: Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, 1988.

23. A. E. Charlesworth. An approach to scientific array processing: The architectural design of the AP-120B/FPS-164 family. *IEEE Computer*, 14(3):18–27, 1981

24. RSX-11M; http://en.wikipedia.org/wiki/RSX-11

25. VAX; http://en.wikipedia.org/wiki/VAX

26. VMS; http://en.wikipedia.org/wiki/Virtual_Memory_System

27. J. Nitsche and J. Waters; "The Detection of ASW Aircraft by Their Underwater Acoustic Signatures-A Theoretical and Experimental Investigation (U)," *Proc. 29th Navy Symp. Underwater Acoustics*, Vol. 2, U.S. Navy Office of Naval Research, Nov. 1972, pp. 111-128. Confidential.

28. J. Barger, J. R. Nitsche, and D. Sachs, "Sound Transmission through the Sea Surface," *J. Acoust. Soc.of America*, Nov. 1976, Supplement No. 1, Fall 1976, pp. S35 (A).

29. J. R. Nitsche, J. O'Connor, and J. Bigler, "A Comparative Estimate of the Underwater Detectability of the P-3 and S-3 Aircraft (U)," *Proc 32nd Navy Symp. Underwater Acoustics*, vol. 2, U.S. Navy Office of Naval Research, Nov. 1978, pp. 407418. Confidential.

30. D. Steele et al., "AUSEX Brassboard Processing System," BBN Report 4497, Nov. 1980.

31. The BQQ-5 was the designation of the sonar system installed on U.S. Attack Submarines at that time. The TB-29 towed-line array used was part of the BQQ-5 system.

32. H. Briscoe and E. Starr, "Post-Test Analysis Detection Results," BBN Report 4496, Nov. 1980.

33. In 1986 the Walker brothers (U.S. Navy personnel) were convicted as spies for giving U.S. Navy classified information to the Soviets. Many in the ASW community speculated that from this information over the years the Soviets learned of the acoustic vulnerability of their submarines, and were then motivated to invest in quieting their newer submarines, such as the Akula class. See T. Parrish, *The Submarine: A History*, Viking, 2004, p. 488.

34. The underwater part of the ARIADNE project was canceled during this time due to technical difficulties. The BBN ARIADNE shore signal and information processing design was adapted to be an advance development model for the exploratory phase of the Fixed Distributed System, the designation of the replacement to the SOSUS system.

35. VME Bus; http://www.lecroy.com/lrs/appnotes/introvme/introvme.htm

36. Sky Computers; www.skycomputers.com

37. Mercury Computer Systems; http://www.mc.com/

38. ADA Programming Language http://en.wikipedia.org/wiki/Ada_programming_language#History; this requirement was relaxed during the program.

39. Bennedict, John R. "The Unraveling and Revitalization of U.S. Navy Antisubmarine Warfare"; www.nwc.navy.mil/press/2005/spring/art4-sp05.htm

Chapter 11

DataProbe and ADCAP

Thomas Fortmann

In the early 1980s, development of the Navy's first microprocessor-based torpedo — the Mark 48 ADCAP — involved hundreds of in-water tests, each producing 100 times more recorded data than any previous underwater weapon system. BBN created state-of-the-art software, using the latest computer and display technology, to provide Navy engineers with unprecedented interactive capabilities: random access to the data, powerful graphical analysis, and programmability to automate repetitive analysis procedures. Called DataProbe, it soon revolutionized system testing in all the Navy's torpedo programs, obsoleted several archaic and expensive batch-processing programs, and saved the Navy $25M in the ADCAP program alone. Designed to process all manner of recorded data and send output to any graphics device, DataProbe was later adopted by the F-14 and other aircraft and missile test programs. A commercial version was developed and sold to military and civilian customers for a wide variety of applications. Further innovations included real-time data access and display, turnkey hardware test sets programmed on top of DataProbe, and artificial intelligence in the form of an expert system that took over tedious failure analysis tasks from human analysts. DataProbe and ADCAP brought $35M of contract revenue and commercial sales into BBN over a 16-year period and employed 110 BBNers in one capacity or another. The software is still in daily use, 30 years after its conception.

11.1 Torpedo data analysis

In the summer of 1980, Steve Milligan and Tom Fortmann visited the Naval Underwater Systems Center (NUSC) in Newport, RI to learn about the Mark 48 ADCAP (ADvanced CAPability) torpedo program (Figure 11.1†). The electronic subsystems of the venerable Mark 48 — a classic collection of analog circuits and digital logic — were being replaced with a radical (for the time) design involving seven heterogeneous microprocessors all communicating with each other on a bus. The prime contractor was Hughes Aircraft Company in Buena Park, California, and NUSC was the Navy's Technical Direction Agent overseeing the project.

Data analysis for testing the Mark 48 had always been done by synchronously sampling a modest number of signals, recording them on tape, and printing them out on long, unannotated strip charts with the horizontal axis representing time. Expert analysts would then roll them out (down a long hallway in some cases), pore over them with classified transparent overlays containing the axis scales, and try to understand what the torpedo system had done during a test run.

†Figures 11.1, 11.2, 11.6, 11.7 and 11.8, and the Morris figure on page 248 are posted in color on the book's website, mentioned in the preface.

Figure 11.1 ADCAP in action vs. ex-*Torrens*.

The plan for ADCAP data was to capture all asynchronous bus traffic (messages) among the seven processors on a massive (for the time) 14-channel instrumentation tape recorder: approximately 2.5 gigabytes with thousands of variables in a single test run, or about 100 times more data than in any previous underwater weapon. The data streams recorded in ADCAP's Test and Evaluation phase are shown in Table 11.1.

Table 11.1 Data stream abbreviations and names

ADP: Autopilot Data Processor	SP: Signal Processor
TDP: Tactical Data Processor	RCV: Receiver (sonar)
I&E: Instrumentation & Exercise	WB: wideband (sonar)

PFP: Portable Firing Panel (separate launch-control device)
3D: Test range tracking data (positions of torpedo and target)

We pointed out, and NUSC agreed, that the old analysis model of rolling out strip charts was not going to cut it.

BBN proposed an interactive system, christened *DataProbe*, wherein an analyst would sit at a computer terminal and call up plots and tabulations of various combinations of variables over selected time segments. The asynchronous data and very different nature of the information in each processor would make for a complex and difficult implementation.

The system specifications, however, had not anticipated these issues and Hughes was unwilling — without a change of scope — to do more than duplicate the strip-chart model. With NUSC's encouragement, BBN bid an interactive system for the data reduction subcontract and lost on cost. The winners, McDonnell-Douglas and Telos, proceeded to develop a batch-processing system called ADRP (ADCAP Data Reduction Program) that ran all night to produce a huge pile of line-printer output with carbon-paper copies containing predefined plots and tabulations of perhaps 15 percent of the data. The other 85 percent (accessible in DataProbe) were ignored by ADRP.

NUSC's point man for data analysis, Mike Altschuler, convinced his colleagues that the Navy needed an independent, interactive data analysis capability and initiated a delivery-order contract with BBN in March 1981 to get it started. DataProbe version 0, code-named "Altschuler's Revenge," was demonstrated in Newport in June. John Means became NUSC's manager for data analysis later that year and assumed the role of godfather to DataProbe for the next two decades.

DataProbe was running in a limited form in time for the first ADCAP in-water tests; its capabilities were expanded and improved rapidly as analysts used it and demanded more. Addition of command scripts reduced the initial turnaround time for an in-water test from all day to less than an hour. In retrospect, it became clear to NUSC management that the ADCAP Test and Evaluation schedule could not have been met without such a capability. The power of interactive processing and display combined with programmability revolutionized data analysis in the torpedo community.

Hughes spent $23M of the government's money on ADRP and then abandoned it in March of 1983. That event inspired the DataProbe motto

Orchides forum trahite; cordes et mentes veniant.

which adorns the now-famous DataProbe beer mug — the first non-coffee receptacle to be issued by a BBN project (Figure 11.2).

Program manager John Means proved to be a shrewd and tireless champion of DataProbe, and NUSC issued delivery orders totaling $20M to BBN from 1981-1996. About a quarter of that paid for development, configuration control, testing, maintenance, and documentation of the core software; the balance was for ADCAP support activities, application software written on top of DataProbe, and extensions to other weapon systems. A 1994 Navy document justifying conversion to the commercial product estimated that the $8M spent on DataProbe and FDRS (see below) had saved the ADCAP program $32M.

11.2 Technical challenges

Extracting data from what amounted to a mammoth, asynchronous telemetry stream was a prodigious task. Some data (e.g. from the sonar) were sampled synchronously at high, fixed rates, but some of the most important data were quite asynchronous, with samples available only when a message (data packet) from the corresponding processor appeared on the bus. Other data took the form of "events" that occurred only occasionally during a test. Correlating samples with time was doubly difficult because a complex set of signals and resets at the beginning of the tape had to be interpreted to determine just where the data recording began.

The data were transcribed from the 14-channel instrumentation tape to one or more 9-track computer-compatible tapes[1] for each on-board processor — often a dozen or more tapes from a single test run. Today, when a full 2.5-gigabyte ADCAP data set would fit on a postage-stamp-size memory card in a digital camera, the whole process seems quaint. But in 1980 a state-of-the-art 30-megabyte removable disk drive — the size of a small refrigerator — represented a major budget item, and one megabyte of RAM maxed out a VAX computer. Thus access to the data would necessarily involve mounting and dismounting many tapes and be inherently sequential.

Demand for access to test data was expected to be high. Multiple experts would be waiting impatiently to analyze each test run (one or more per day at the height of the test and evaluation phase), in order to modify the torpedo and plan the next test. Moreover, they would need to view the data in a variety of ways and modify their analyses on the fly based on what they found. They openly ridiculed Hughes' overnight batch data reduction program (ADRP) with its inflexible mountains of mostly irrelevant line-printer output.

Morris the Cat

NUSC personnel were sometimes nonplussed by BBN, which was very different from the contractors they were accustomed to. John Means reminisces:

> Most contractors would do anything for the almighty dollar. When you read their proposals they could do anything for anybody, anyplace, at anytime. The most successful companies knew what their forte was. BBN was in that category. Your engineers knew what they were good at and consequently they excelled at it!
>
> At the onset of our relationship, I spent a lot of time justifying your man-year rate. As time went on, that issue went away because all learned that if you spent $1 at BBN you got $3 in return. The BBN focus was the product, not the time clock. You folks put your hearts and souls into DataProbe!

At one point, NUSC needed modifications to some old COBOL programs and Means asked BBN to take it on. We declined, explaining that if we asked any of our programmers to learn COBOL they might resign. Means was astonished that a contractor would turn down billable hours and from then on we were known as "Morris the Cat" (Morris starred in television ads at the time, turning up his nose at anything other than 9Lives brand cat food).

Nevertheless, BBN's single-minded focus and ability to solve the most important and challenging data problems were key factors in the ADCAP program's success, and DataProbe saved the Navy a lot of money as well. Means reminds us:

> Never forget that our job was to build the best damn torpedo for the US Navy and that the "Probe" was an integral part of achieving that goal. I remember when the data reduction for the first in-water runs turned into a nightmare and you guys saved the day with a "rump" version of Probe.

11.3 Data caching and tape slavery

Milligan and Fortmann sketched out a design using state-of-the-art (for the time) equipment: a VAX 11/780 computer with VMS operating system from Digital Equipment Corporation (DEC) and a Tektronix 4014 storage-tube terminal as the output device. Displaying information on the screen under interactive operator control would prove to be relatively straightforward compared to dealing with the labyrinthine torpedo data streams. Programming was done in RATFOR (RATional FORtran), Bell Labs' precursor of the C language that preprocessed C control structures (but unfortunately not C data structures) into FORTRAN.

To provide random, shared access within the hardware limitations at that time,

Figure 11.2 DataProbe beer mug with logo and motto.

Milligan conceived and implemented a software architecture that pushed the hardware and operating system to their limits and was nothing short of brilliant. The *DataCache*, sketched in Figure 11.3, utilizes three levels of caching, both on disk and in random-access memory (RAM):

- Multiple copies of an autonomous program called a *Tape Slave* are run, one controlling each active tape drive. The Tape Slave deals with physical/logical records, the launch sequence, time, and other idiosyncrasies, storing recently read physical records on disk to minimize thrashing.[2] It responds to data requests from other processes by placing physical records in the common-memory record cache, along with pointers and time stamps for the requested logical records contained therein.
- The *common-memory cache* is shared among all concurrent DataProbe users. It responds to requests of the form "get(variable, t_0, t_1)" by using the Database Dictionary (see below) to unpack samples of individual variables from the logical records.
- A private *variable cache* is maintained for each user, permitting multiple analysis procedures to be conducted on the same set of recorded variables. It also allows users to manipulate, modify, and redisplay plots without repeating data retrieval. Except for the densely and synchronously sampled sonar data, every sample is paired with a time stamp.

The common-memory cache was implemented using VMS shared memory, with VMS mailboxes providing synchronization signals across the various processes.

Small data streams such as PFP and 3D were stored directly in disk files. Eventually disk space became plentiful enough to hold copies of the previously tape-resident data, but independent Tape Slaves are still critical for dealing with the disparate data streams and time anomalies.

Milligan also devised a means of representing the plethora of recording modes in a *Data Dictionary* that could be modified[3] as new modes or errors were encountered,

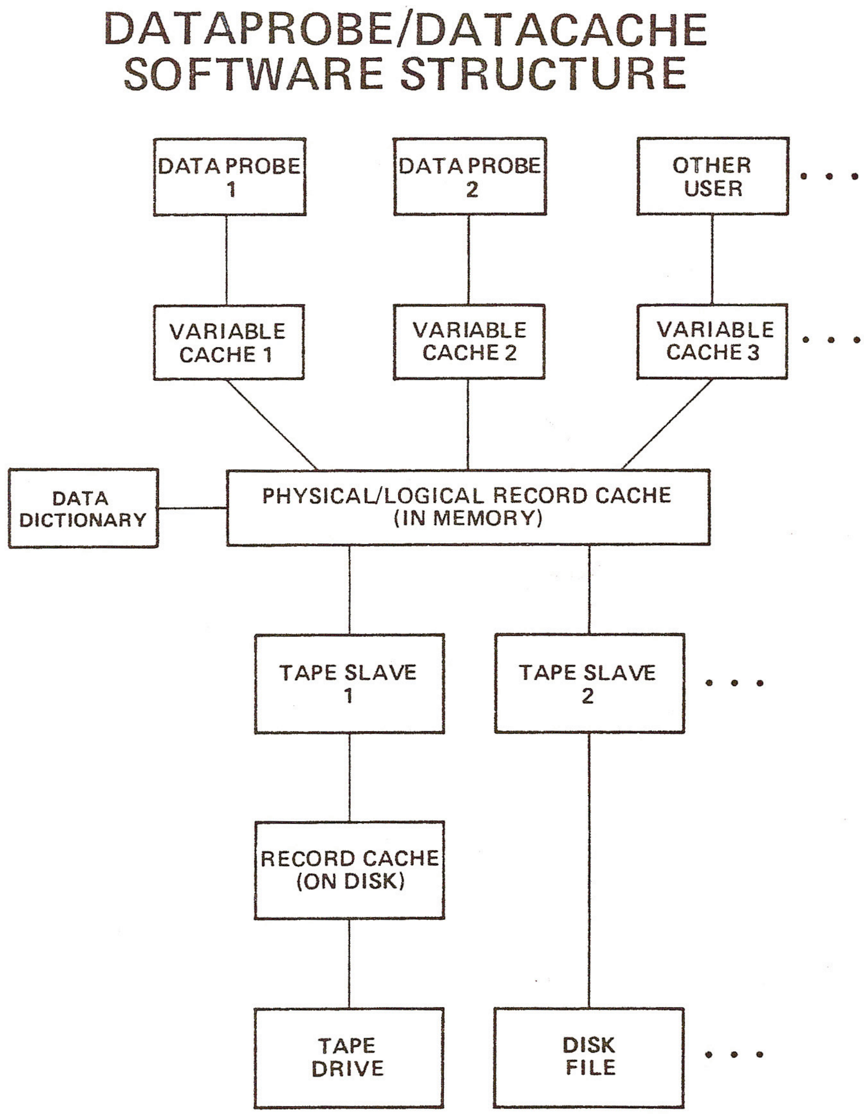


FIG. 3. DATAPROBE/DATACACHE SOFTWARE STRUCTURE

12

Figure 11.3 DataCache structure. (From DataProbe marketing blurb, circa 1983.)

Twenty-eight years later, Steve Milligan reflects

I always thought the "great breakthrough" was the notion that multiple analysts could have random access and share an intrinsically sequential and non-shareable resource (multiple tape drives). The sequential, non-shareable nature of tape drives is why Hughes was stuck on batch analysis. They couldn't imagine doing anything except read tapes front to back with a dedicated program. The notion of the DataCache (and the Tape Slaves feeding it) allowed both shared and random access by multiple analysts simultaneously. This and the Data Dictionary made everything else possible. In particular, decoding the launch sequence peculiarities was impossible without random access. Of course, eventually disk storage caught up with tape capacity and one could have just copied everything to disk, but back at the beginning there was more data than anyone had ever considered for random access.

without changing the underlying program. This proved to be another critical innovation, keeping the unpacking and display of data independent of one another and enabling rapid adaptation of DataProbe to new data sources and recording systems.

A few years later, Ben Dubrovsky expanded the data-dictionary approach to create the *Flexible File Server*, which enabled user-configurable access to an entire data set on disk or tape: physical and logical records; record lengths, IDs, and time tags; as well as the individual variables stored within. This capability was critical to commercial success because it enabled a support engineer to quickly connect DataProbe to a customer's data during a single sales call. It was later modified to handle real-time data, with the simple artifice of polling a data source and adding data to a growing file.

The beauty of this architecture — autonomous tape control, multiple levels of caching, and a program-independent table of record structures — was that analysts had only to think about specific variables of interest; all details of the data extraction process were conveniently invisible.

11.4 User interface and data display

At the same time, Fortmann and new hire Jim Arnold were implementing the more visible components of the software, basing the user interface on a command-line-interpretation library called COMAND. With roots in the TENEX operating system (Chapter 21, page 523) and the BBN Speech Group (see sidebar), COMAND was developed in earlier sonar signal processing and tracking projects and extended/refined for DataProbe.

The user controlled DataProbe (and the earlier sonar programs) by means of a novel command-line interface with automated command recognition: typing "?" would display all available choices, "escape" would fill in a command or subcommand, and "noise words" indicated what input was expected next. Jeff Berliner had designed a clever COMAND-based utility called PARCHG (for PARameter CHanGe) that displayed a set of numerical and other parameters — for example, to configure a time plot — and allowed the user to change any of them before continuing.

DataProbe graphics were built on PLIB, a remarkably versatile device-independent library of graphics subroutines that originated in the same sonar projects. Tom Hammel, collaborating with Berliner and others, designed PLIB so that an application programmer could concentrate on data and ignore pixels. This was accomplished by maintaining two internally mapped entities: "data space" and "pixel space." Once the programmer established a mapping between the two, the application program could define graphs

Evolution of user interfaces and reusable software

BBN's TENEX operating system, introduced in 1970 for the DEC PDP-10, was revolutionary in several ways, perhaps the least famous of which was its interactive user interface. Other systems of that era, including UNIX, allowed users to type commands, subcommands, and parameters. TENEX took that model to the next level with meaningful command words (like "DIRECTORY" instead of "ls") and explanatory "noise words" to indicate what user input was expected next. To minimize keystrokes, it allowed the user to type abbreviations, "?" for a list of available commands, and "ESCAPE" to fill in commands and noise words. The same features later appeared in the UNIX shell "tcsh."

John Makhoul, Jerry Wolf, and Rich Schwartz of BBN's speech group created a FORTRAN-callable library, LIB10, that gave application programmers access to TENEX system calls, including command recognition/completion. Jeff Berliner and Dick Estrada, with help from Tom Hammel and Doug Cochran, adopted and extended LIB10 and its successors, COMAND and PARCHG, for a variety of sonar signal processing and tracking projects at the ARPA Research Center (ARC) at Moffett Field, CA. Hammel later merged and redesigned COMAND and PARCHG, adding the ability to read command scripts.

It was this large base of versatile, reusable software (also including PLIB) that enabled us to quickly prototype and demonstrate a powerful interactive tool like DataProbe and convince NUSC to invest in its further development. Without such a base, the two-decade-long project would have been stillborn due to prohibitive cost.

Ron Scott, who moved back and forth between BBN and graduate school during those years, offers this perspective:

> From my point of view, the difference between software development in 1977 and 1979 was striking. In 1977 we were developing software to solve a particular problem. By 1979 we were able to think about developing software components that could be used for our particular problem, and reused in the future. I attribute this partly to the use of COMAND and PARCHG, partly to the use of RATFOR (which let us abstract up a level from FORTRAN), but also to the critical mass of smart software engineers we now had to think about these issues.

COMAND and PARCHG optimized the user interface for paper terminals, text-only screens, and the static Tektronix 4014 display (one could draw complex text and graphics but the entire screen had to be erased at once). True graphical user interfaces (GUIs) began to appear a few years later, as soon as dynamic bit-mapped display technology became available.

and plot data without regard to pixels, resulting in clean, device-independent graphics code.[4]

PLIB's device independence also enabled DataProbe output to be directed to a variety of displays and plotters in addition to the venerable Tektronix 4014.

Early releases of DataProbe could plot (or tabulate) one or more variables vs. time, on user-specified intervals and scales, dealing with asynchronous samples and extrapolating where necessary across gaps in the data. The user could select and interactively label a data point or display the mean and variance over a selected interval, as shown in Figure 11.4. Discrete events were recognized and displayed appropriately. An X-vs-Y mode was available to plot torpedo and target trajectories.

Navy torpedo analysts — accustomed over decades to rolling out strip charts — were astonished and delighted to have rapid, interactive, random access to the data. Their appetites whetted, they soon clamored for more sophisticated features such as searches for data points that exceeded specified limits, outlier removal, correlations, smoothing functions, spectral plots, histograms, and "3-D" style graphics. Paper strip charts were

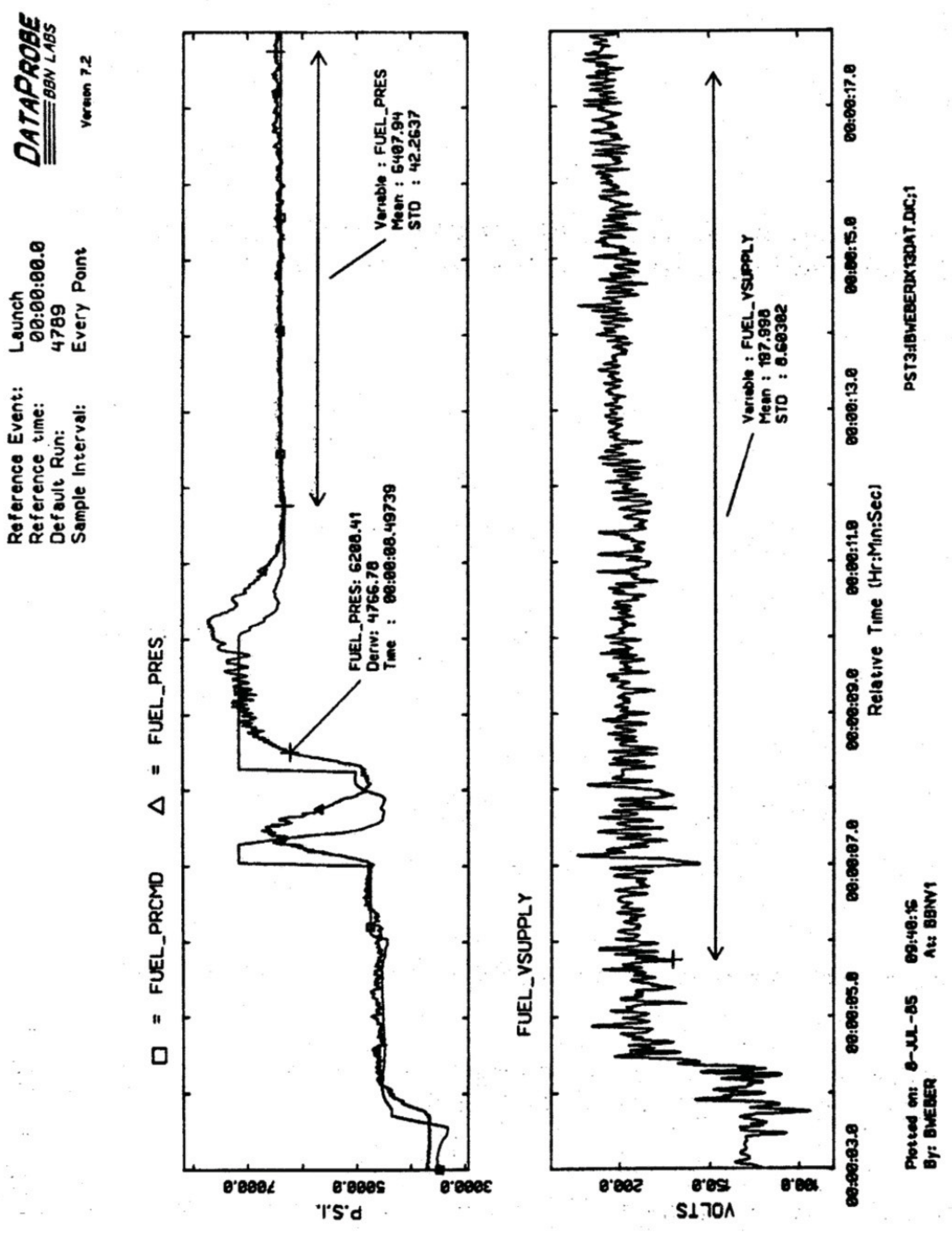


Figure 11.4 Time plot with user-generated labels.

soon forgotten as DataProbe's new capabilities revolutionized the world of torpedo data analysis.

BBN responded with frequent software releases that implemented these requests and others from program management to deal with new data types and situations. This somewhat frantic style of work fit into BBN's tradition of "rapid prototyping" software in close collaboration with customers. Indeed, the pace of software changes was so rapid that the official NUSC requirements specification did not appear until eight years later, as BBN Report No. 5891.

Joe Weinstein joined BBN and the project in the fall of 1982, redesigning COMAND/PARCHG for a third time to handle large menus with potentially thousands of variable names and to support a complete macro programming language. This gave users the ability to automate repetitive procedures, perform and display mathematical computations on the data variables, and — most importantly — make conditional decisions based upon the contents of the data under analysis. Gary Rucinski later worked with him to develop "external functions," whereby analysts could write their own filtering and other algorithms and have them invoked on time series within DataProbe.

Analysts swarmed over the new *automated analysis* features, which multiplied their productivity by eliminating hours of tedious typing and visually scanning plots for specific conditions. Moreover, the introduction of these capabilities fundamentally changed DataProbe's role, turning it into a platform upon which major applications like FDRS, FAES and Mark 661 (see below) could be programmed.

As newer technologies appeared, the generality of the PLIB graphics library allowed DataProbe output to appear on a variety of graphics terminals and hard-copy plotters. DEC bundled a high-resolution screen with a miniature VAX, calling it a VAXstation. BBN, not missing a beat, added DataProbe to the bundle and sold a number of hardware/software packages as "ProbeStations." Ben Dubrovsky developed "Distributed DataProbe," allowing the control and analysis portions of the software to run on multiple workstations connected over a network to data collected on a bigger machine.

Workstation graphics supplanted the static Tektronix displays, enabling a graphical user interface (GUI) and more dynamic, real-time-oriented displays. A variety of animated gauges, dials, and scrolling time plots were demonstrated at DECWorld in Cannes in 1988, using flight test data from Dassault Aviation.

11.5 Other applications and platforms

The ADCAP/DataProbe project grew steadily, adding staff, customers, and tasks. We gave training courses, attended meetings, visited test ranges, wrote custom programs to deal with exotic data types, and supported NUSC's engineers and analysts in a variety of ways. Veteran BBN engineer Howard Briscoe was particularly adept at helping NUSC personnel address system engineering issues. A secure laboratory, complete with its own VAX and peripherals, was constructed so that we could accept and process test data classified up to SECRET level.

Once DataProbe's success in the ADCAP program became known — primarily through John Means' internal marketing — other Navy programs became interested. One of the first — requested by NUSC personnel — was an extension to process data from the original Mark 48 torpedo.

The Naval Undersea Warfare Engineering Station (NUWES) in Keyport, Washington requested adaptations for the Mark 46 and Mark 50 lightweight (airborne) torpedoes and for other systems under their testing purview such as the Mark 40 (ADMATT) and Mark 69 simulated targets and countermeasures. Sam McKeel, Paul Hughes, Steve Stuart, Bill Penner, Tai Lammi, and Kathy Curry — all enthusiastic early adopters —

leveraged DataProbe to modernize and standardize analysis procedures throughout NUWES using a single tool. Stuart reported that Mark 50 proofing analysis was reduced from a 4-6-week process to a one-day quick-look and a full report within a week.

The Naval Ocean Systems Center (NOSC) in San Diego also got involved with the Mark 54 and Mark 46 torpedoes.

When a commercial version of the product became available, a variety of military and civilian customers adapted the product to their needs (see section below). Other BBN projects, most notably SIMNET (Chapter 20), experimented with DataProbe for analyzing a variety of test data. Even BBN's office in Edinburgh, Scotland got an on-site training course and applied DataProbe to projects involving "smart" process control.

Peter Dick led a growing software development team that included Joe Weinstein, Gary Rucinski, Bill Russell, Lisa Kempler, Lucy Karis, Tom Welch, Ben Dubrovsky, Dan Sullivan, Muriel Prentice, and Jenifer Tidwell. They added features, worked with users to identify and implement new functionality, streamlined the code, and learned to pronounce words like "configuration management" and "documentation." The ceaseless task of fixing bugs was attacked with aplomb; Karis recalls fondly her title of "Bug Queen" and Dick once articulated the "Dense Bug Theorem" (between any two DataProbe bugs there exists another bug). Russell and Rucinski added spectral analysis capabilities, auto-and cross-correlations, histograms, color spectrograms, and the product's signature 3-D display (in the background of Figure 11.8 below).

Rucinski, Russell, Jeanne Secunda, Tom Lynch, and Kathy Timko provided technical support and training courses to Navy analysts and engineers, including development of the Performance Analysis System (PAS), its predecessor, the QuickView tactical summary, and other special-purpose software. PAS consisted of automated command scripts that used DataProbe to sift data from an ADCAP test run, detect certain milestones, and extract performance measures at the times of the milestones, collecting a small data set for each run. These data sets were accumulated into a multi-test database in RS/1, where statistical analyses could be performed to evaluate performance in the aggregate.

Stellar contractual/financial/administrative support was provided by Pat Falcone, Cathy Corso, Susan Bendott, and the late Laurie Goodman.

BBN/Newport personnel Brian Palmer, Matt Hefferman, and Miguel Oyarzun provided on-site support and software maintenance, and later adapted DataProbe for use in NUSC's Weapons Analysis Facility (WAF — see below).

Ports to other operating systems followed a few years later, most notably to UNIX with help from software *maestro* Fred Webb. After the product was resident at BBN/Domain, it was also ported to OpenVMS and Windows.

Many ingenious adaptations and extensions of DataProbe — and its ancestors in the earlier sonar projects — all contributed to the Department 47 motto, prominently displayed on the third floor of 10 Moulton Street and later in cavernous 70 Fawcett Street:

Our software can almost do almost anything.

11.6 FDRS and Mark 661

The ADCAP torpedo test and evaluation program, greatly enhanced by the serendipitous emergence of DataProbe, proceeded on schedule from its advanced-development phase into full-scale engineering development. Its ultimate deployment to the fleet would involve routine testing on a smaller scale, and the Navy had allocated $20M for Hughes and its subcontractor, McDonnell Douglas, to build a Fleet Data Reduction System (FDRS) for that purpose.

FDRS was a turnkey system to produce a fixed set of standard plots and reports after each fleet test run; interactive failure analysis, when necessary, would take place elsewhere. Because DataProbe by this time was fully programmable, NUSC and BBN observed that FDRS could be implemented on a small VAX using DataProbe command scripts at a fraction of the budgeted cost.

So it was that Jeff Berliner, Gary Rucinski, Jeff Freilich, and Jeff Schutzman took over development of the FDRS software from McDonnell Douglas in late 1983, working under the direction of NUSC's Jim Wasel. They developed, tested, and delivered Release 1.0 in April 1985, integrating it on site in Keyport. It underwent further testing at NUSC's Life Cycle Support Facility and was accepted for fleet use, saving the Navy the lion's share of the previously budgeted $20M.

The new Mark 50 lightweight torpedo, scheduled for deployment about two years after ADCAP, also needed a standalone data reduction system for fleet testing, in this case known as the Mark 661 Test Set. With the recent success of FDRS, it was not difficult to convince the Navy to use the same approach and build it as an application on top of DataProbe running on a small VAX.

Berliner, Peter Dick, Tom Lynch, Nuriel Lapidot, and Doug Brockett procured the hardware and coded the DataProbe application software to implement the Mark 661, integrating it on site in Keyport in the summer of 1989. This was the first time BBN delivered a full, certified, turnkey hardware/software system for use in the Fleet (Figure 11.5). As with FDRS, the Navy realized substantial savings.

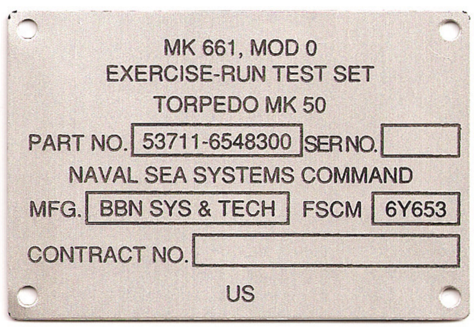


Figure 11.5 Metal tag affixed to MK661 Test Set.

11.7 Point Mugu

A number of aircraft test labs also became interested and purchased copies of DataProbe, most notably the Navy's Pacific Missile Test Center (PMTC) at Point Mugu, California. The SITS[5] Laboratory (Figure 11.6) featured an F-14A airframe with full electronics and radar systems and a large door overlooking the Pacific Ocean where test targets could fly by to exercise the aircraft's radar.[6] SITS program manager Sam Wilson contracted with BBN/LA staff Matt Sneddon and Jose DeBarros to interface DataProbe to his recorded test data. Other PMTC groups saw the tool in the SITS lab and soon adapted it to the EA6B aircraft and AMRAAM and Phoenix missile systems in test labs at Point Mugu and elsewhere.

At around the same time, Wilson was expanding the SITS lab to accommodate the new F-14D aircraft. He procured a BBN Butterfly parallel processing computer (Chapter 21, page 538) to control the real-time testing process and engaged BBN/LA's

Figure 11.6. Radar Intercept Officer (RIO)'s ocean view from F-14D airframe in Point Mugu's SITS Laboratory.

team — expanded to include John Nash, Lynn Omelchenko, Doug Brockett, Anita Hsiung, and program manager John Hough — to build a custom VME-based front end that interfaced it to the airframe's 1553 bus and develop a variety of software. This also made data available for real-time display and analysis during experiments, a big improvement over the post-test data processing in F-14A tests.

Heavy demand for the SITS Lab from a variety of groups on the base meant that scheduling and frequently rescheduling its complicated set of resources required near-constant attention from a dedicated staff member. In a separate project, BBN's AI group designed scheduling software based on a genetic algorithm (GA); it worked so well that the formerly indispensable staff member was able to retire.

11.8 ButterProbe

In addition to the SITS lab at Point Mugu, BBN's Butterfly was the basis for another very compute-intensive hardware-in-the-loop simulation facility at NUSC's Weapons Analysis Facility (WAF) in Newport, RI. Both labs simulated the operational environment of a major weapon system — torpedoes underwater in one case and aircraft flying in the other — in real time, connected to the weapon's inputs and outputs, and exercised the weapon's sensors and computers during realistic test scenarios.

These and other potential applications needed real-time displays of live simulation data as tests were in progress. This led to the idea of extending Distributed DataProbe

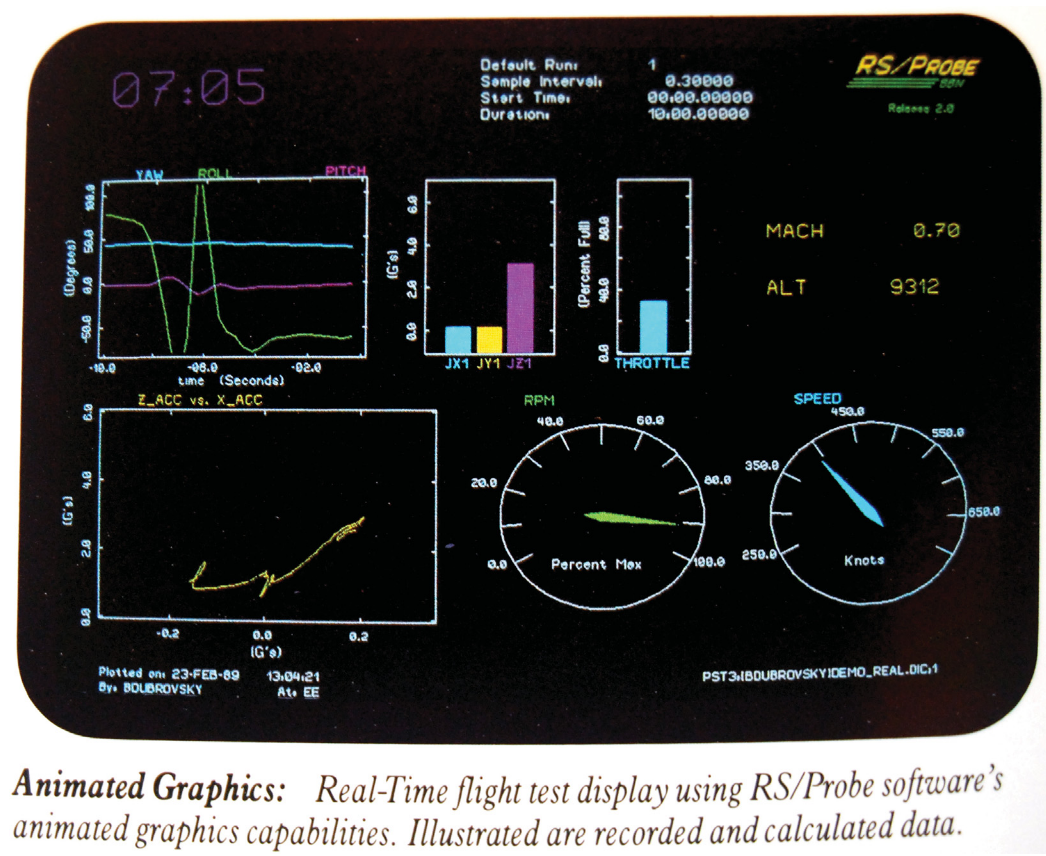


Figure 11.7 Animated display of real-time data.

to incorporate real-time data collection on the Butterfly (or other machine) with data displays on separate computers or workstations.

To accomplish this task, Doug Brockett and Joe Weinstein generalized Steve Milligan's original Tape Slave/DataCache design. They created a real-time, tape-slave-like component in pSOS on PMTC's Butterfly that passed data frames to the *Data Exchange*, a generalization of DataCache on another computer. Dave Cousins and Brian Palmer at BBN/Newport created a similar process for the NUSC Butterfly. It was a challenge to keep TCP/IP from drowning in data, but eventually data rates of a few hundred kilobytes/sec were achieved. Brockett modularized and optimized the Data Exchange code, eventually achieving throughputs of 20,000 frames/sec or about 10 megabytes/sec.

The Data Exchange provided either data frames or, using the Data Dictionary, individual variables to copies of DataProbe on analysts' workstations. A variety of animated gauges, dials, and scrolling time plots were added to DataProbe to display the real-time data (Figure 11.7). In addition, Brockett collaborated with Anita Hsiung to create DataProbe's first graphical user interface (GUI) as a front end to the command-line interpreter (COMAND); this served as a prototype for the later commercial GUI.

This marriage of DataProbe and the Butterfly led, perhaps inevitably, to the memorable nickname *ButterProbe*.

11.9 FAES and patterns

DataProbe first enabled interactive analysis; later, command scripts made it programmable and led to a host of applications written by both developers and users. The largest of these were FDRS and MK661, which produced standard plots and tables after routine fleet torpedo tests. Human analysts scanned those outputs, using a "rule book" of previously identified shapes and patterns in the time plots that might indicate failure.

BBN's experience with artificial intelligence (AI), especially expert systems, soon led to the suggestion that the failure analysts' tedious tasks might be done faster and better with an expert system. In late 1986, Jon Delatizky and Jeff Berliner worked with the late Ken Anderson to devise a "syntax" of signal elements, parameterizations of those elements, and algorithms for parsing them. The key innovation was that the expert system could then reason about both qualitative shape characteristics ("glitchy flat" was a favorite) and quantitative characteristics of those shapes.

A prototype system, dubbed the *Failure Analysis Expert System* (FAES), was built using Steve Jeffreys' communications conduit between DataProbe and a Symbolics Lisp Machine, with Jeff Morrill developing the bulk of the parser and expert system code. Following a successful proof of concept, Tom Lynch and Karl Haberl led a major proposal effort, the Navy funded the project, and Mike Duarte became the primary stakeholder and champion at NUSC. Morrill continued to enhance the code while Delatizky worked with NUSC expert Dan Bowlus to perform the knowledge engineering, creating syntactic descriptions of signal shapes and rules that matched the Navy's interpretation manual. The final version was delivered on Sun workstations running Franz Common Lisp, communicating with DataProbe on an old, slow FDRS VAX.

≈ ≈

FAES opened several other doors. It turned out that using syntactic pattern recognition to analyze the signature of, say, container pressure as a valve opens and closes, has an interesting variety of commercial applications. Many electromechanical systems have characteristic wave patterns that fit no mathematical model but have an expected qualitative shape that is easy to describe geometrically. A system that can break down large amounts of noisy time series data into high-level descriptive patterns or shapes, in real time, proved to be widely useful to a broad range of applications beyond torpedoes and defense applications. How quickly does the valve close? Is the valve degrading over time? Is there an indication of a pressure leak?

The technology developed for FAES was refined, enabled to operate with or without a DataProbe front end, and unveiled as a new product named BBN/Patterns in 1994. Lockheed Martin used BBN/Patterns to analyze telemetry from an Atlas Centaur rocket sitting on a launch pad fully fueled, where the data must be continually analyzed in real time for a potential catastrophic failure which could cause the liquid oxygen to explode. Intel Corporation also became a major customer: BBN/Patterns was used[7] for many years in nearly all its Pentium fabrication plants worldwide to monitor the quality of its manufacturing processes, raising alarms when things seem to go awry.

11.10 Commercial sales

DataProbe was conceived from the beginning as a product with broad utility well beyond the world of torpedoes. The first commercial sale was made to Grumman Data Systems for flight testing in 1985. Sandy Fidell and Tom Fortmann published the first DataProbe article in *Hardcopy* magazine later that year. Articles followed in *Defense Electronics*

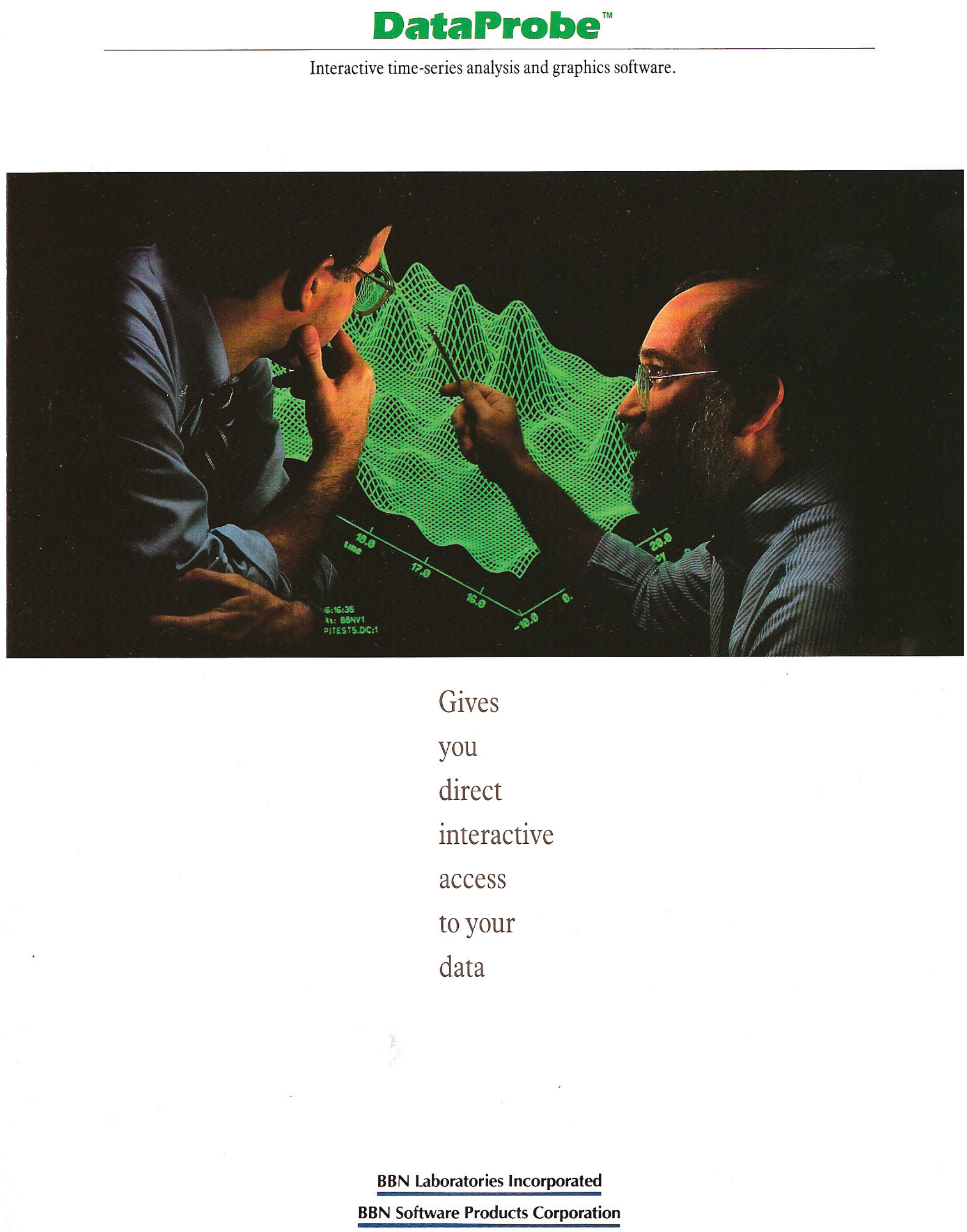


Figure 11.8 First DataProbe brochure, 1986, starring Jeff Berliner.

and *DEC Professional* magazines, a glossy color brochure (Figure 11.8) appeared, a newsletter was published, and DataProbe hit the trade-show circuit with a booth at the International Telemetry Conference in Las Vegas in 1986, organized by BBN Labs' marketing communications manager and head cheerleader Donna Lane.

After commercial sales to PMTC (see section 11.7) and to General Motors for driver simulation, BBN Software Products Corporation (SPC) took on sales responsibility in 1987 and appointed Fred Kern to be DataProbe sales manager. Pete Moss transferred to SPC later that year to do sales support, followed by Ben Dubrovsky, whose *Flexible File Server* (described above) paved the way to closing numerous sales. Moss later became product manager, with Lisa Kempler and Lucy Karis transferring to SPC in 1989 to do development and some customer support. Mindy Garber from BBN/ACI, Mark Ross, Mark Avenmarg, and Chris Chiapella joined to provide training, documentation, and customer support.

Kern's solitary efforts soon expanded to include sales staff around the globe: Syd Schips, George Danielson, Linda Bernardi, Tom Finn, Nadine Nastasi, Laura Hyde, Steve Scott, Lori Waldron, Jan Willem deGraaf in Holland, Darron Passlow in Australia, Yasuo Komoto and John Scandurra in Japan, and sales manager Rich Schembor.

In 1988 the commercial version was renamed *RS/Probe* for consistency with SPC's flagship product, RS/1, but a year later the name changed again to *BBN/Probe*. NUSC's version retained the name DataProbe. Other customers included those listed in Table 11.2.

Addition of a modern graphical user interface (GUI), along with ports to UNIX, OpenVMS, and Windows, enhanced the product's attractiveness and expanded the potential customer base.

Sales peaked at $2.3M in 1990 and in 1991 SPC transferred the product and staff back to BBN Labs. Commercial sales continued at $1.5-2M per year, with a total just over $12M for FY1988-94 in a wide variety of applications. During this period the Navy's FAES technology — a DataProbe-based expert system for detecting failure modes — was turned into a commercial product called BBN/Patterns (see section above). Patterns' best customers were Lockheed Martin for Atlas Centaur rocket launches and Intel Corporation for monitoring manufacturing quality.

In 1994, weary of maintaining two parallel products, BBN offered and NUSC accepted a cost-free license to use the commercial BBN/Probe in place of DataProbe. This provided them access to UNIX, OpenVMS, and Windows platforms as well as enhancements like the Flexible File Server and GUI. The license, negotiated by Connie Garand on behalf of the Navy, also gave them rights to the source code at no cost if BBN or its successors ever discontinued support of the product.

Later that year BBN/Probe, BBN/Patterns, and associated staff led by Tom Lynch were once again transferred to SPC, by then renamed BBN Domain. In 1996 the subsidiary spun out of BBN to become Domain Solutions and then Domain Manufacturing.

In 1999 Domain and all its products were acquired by Brooks Automation of Chelmsford, MA. In 2000 Brooks discontinued development and support of BBN/Probe, offering to sell the source code at a high price to customers who wanted to continue using the product. They were more than a little surprised when the Navy exercised its option to obtain the source code for free.

11.11 Epilogue

Perhaps the best testament to the utility of DataProbe and the ingenuity of those who created and nurtured it is that this software, conceived and demonstrated in 1980 on the first VAX and a now-obsolete storage-phosphor display, is still in active use today, running on a variety of computers and operating systems.[8] Key applications like FDRS and the MK661 Test Set continue to run on top of it, but FAES was eventually discontinued due to lack of funding.

NUSC, renamed the more memorable NAVUNSEAWARCENDIV NEWPORT, now maintains all the software in house, having obtained the source code when commercial

Table 11.2 Partial list of customers of *BBN/Probe*

Army Missile Command Huntsville, AL	Holloman AFB Alamogordo, NM	Northrop Grumman Pico Rivera, CA
Bendix/King Avionics Fort Lauderdale, FL	Honeywell IAC Phoenix, AZ	Pacific Missile Test Center Point Mugu, CA
Boeing Commercial Aircraft Seattle, WA	Hughes Satellite Group El Segundo, CA	Raytheon Corporation Multiple locations
Boeing Military Aircraft Wichita, KS	Hunter Liggett Army Base Salinas, CA	Sikorsky Helicopter Bridgeport, CT
Dassault Aviation Toulouse, France	Il Moro di Venezia America's Cup Challenger	Swiss Air Force Payerne, Switzerland
Edwards AFB Antelope Valley, CA	Kirtland AFB Albuquerque, NM	SYSTRAN Corporation San Diego, CA
Eglin AFB Valparaiso, FL	Lawrence National Lab Livermore, CA	Taiwan Semiconductor Mfg Hsinchu, Taiwan
Exxon Corporation Baton Rouge, LA	Loral Corporation Palo Alto, CA	TRW Systems Group Redondo Beach, CA
General Dynamics San Diego, CA	McDonnell Douglas St. Louis, MO	UCLA Physiology Dept Los Angeles, CA
General Dynamics Fort Worth, TX	NASA Ames Research Ctr Mountain View, CA	Wright-Patterson AFB Dayton, OH
General Motors Detroit, MI	Naval Air Test Center Patuxent River, MD	Yuma Proving Ground Yuma, AZ
Grumman Data Systems Bethpage, NY	Naval Air Weapons Station China Lake, CA	

support was discontinued. NUWES, now NAVUNSEAWARCENDIV KEYPORT, uses DataProbe daily on Windows PCs to analyze data from both heavyweight and lightweight torpedo tests.

Other military customers purchased the source code in order to maintain the product. Members of the former BBN/LA office, now in private consulting practices, use DataProbe for a variety of airport-noise environmental impact studies and acoustic analyses.

Funding associated with ADCAP, DataProbe, and their offspring came from dozens of sources, the exact total of which is lost to posterity. An educated guess is that from 1981 to 1996 BBN received nearly $20M in Navy delivery orders and other military funding plus another $16-17M in commercial sales, for a grand total of perhaps $35M over 16 years. Domain Manufacturing and Brooks Automation continued to derive revenue after the 1996 spin-out.

Perhaps the best measure of DataProbe and ADCAP's impact on the company is the number of BBNers who were involved in the project at some time in some capacity. Below is a (probably incomplete) list.

≈ ≈

All in all, it was a great run — thanks to everyone for jobs well done, and especially for the memories!

Notes

1. One foot diameter, 6250 CPI, 140 megabytes.

2. “Thrashing” refers to excessive back-and-forth tape motion.

3. By means of a separate, interactive, Data Dictionary editor program.

4. PLIB’s unit of screen measurement (a sort of pseudo-inch) was christened the *mucci*, in honor of signal processing guru Ron Mucci. Unfortunately, it never attained the local celebrity of the *smoot*.

5. System Integration Test Stand

6. Lab staff were once reprimanded for zapping seagulls on their lunch break.

7. It may still be used — we have been unable to find out.

8. One attempt to use VAX/11-780 emulation software foundered on security concerns because it had been written by Russian programmers.

Appendix: BBN personnel involved in the DataProbe and ADCAP

Ken Anderson
Jim Arnold
Pam Aumann
Mark Avenmarg
Donna Belleau
Susan Bendott
Marc Berkowitz
Jeff Berliner
Linda Bernardi
Howie Briscoe
Doug Brockett
Ed Campbell
Chris Chiapella
Doug Cochran
Cathy Corso
Lynn Cosell
Dave Cousins
George Danielson
Peter Dick
Jose DeBarros
Jon Delatizky
Ben Dubrovsky
Gary Dworman
Tom Dyer
Laura Eberhard
Tom Elliott
Tad Elmer
Dick Estrada
Pat Falcone
Sandy Fidell
Tom Finn
Tom Fortmann
Bobbi Freeman
Jeff Freilich
Bob Gagnon
Mindy Garber
Laurie Goodman
Dan Gordon
Jan Willem deGraaf
Karl Haberl
Tom Hammel
Michael Harris
Matt Hefferman
Muriel Hervey
Dave Hickerson
Paul Horwitz
John Hough
Anita Hsiung
Bill Huggins
Marcy Hunter
Laura Hyde
Steve Jeffreys
Kathie Jones
Lucy Karis
Lisa Kempler
Fred Kern
Yasuo Komoto
Laura Kurland
John Kyratzoglou
Donna Lane
Nuriel Lapidot
Jeanne Lee
Ina Loobeek
Jim Louie
Tom Lynch
Debbie Maloney
Bill Messner
Steve Milligan
Jeff Morrill
Pete Moss
Ron Mucci
John Nash
Nadine Nastasi
Andrea Nidzgorski
Lynne Omelchenko
Miguel Oyarzun
Brian Palmer
Darron Passlow
Gerry Prado
Muriel Prentice
Bob Pyle
Mark Ross
Gary Rucinski
Bill Russell
Karen Sarachik
John Scandurra
Richard Schaffer
Rich Schembor
Jeff Schutzman
Syd Schips
Ron Scott
Steve Scott
Linda Secrist
Jeanne Secunda
Jim Sheerin
Stan Shursky
Matt Sneddon
Michele Starmer
Dan Sullivan
Jenifer Tidwell
Kathy Timko
Lori Waldron
Fred Webb
Barry Weber
Joe Weinstein
Tom Welch
Ann Wells
Emily Wendell
Fred White

Chapter 12

Medical Applications of Computers

Paul Castleman

Early work at BBN involved hospital computer systems, computer aids for the physician's office, data management tools for clinical research, and database and computational support for biomedical research. The work included both development of prototype systems and later deployment of commercially viable software and services. This history also notes some of the challenges of working within an R&D defense-contractor environment and then concludes with lessons learned in developing medical computer applications.

No sooner had the first primitive three-user time-sharing system been demonstrated in 1962 (see Chapter 4) than Bolt Beranek and Newman began working in the medical application of online interactive computing. During the early decades, most such work was conducted by teaching hospitals and medical schools, with some commercial attempts by computer manufacturers — for example, IBM and Digital Equipment Corp. (DEC). However, BBN was one of the first commercial R&D labs to work in this area. Over the years, BBN's work principally involved remote-access medical data handling; little was done in the areas of realtime applications, image processing, or treatment planning.

In the early 1960's BBN's first major initiative was in medical-record applications for **patient care.** By mid-1960 the system was extended to serve investigators doing **clinical research**. Beginning in 1968, BBN began efforts to support scientific **biomedical research**, and then in the late 1970's added a **commercial software-product** activity. Rather than give a chronological project-by-project account, this paper discusses each of these four areas of activity in turn. The **impact of the BBN environment** on these efforts is then discussed, and finally a summary of personal **observations and conclusions** is presented.

12.1 Patient care

Dr. Jordan J. Baruch, an MIT professor of electrical engineering, was one of the first acoustical engineers at BBN. But his interests were difficult to confine. Jordan was the energetic visionary who sold the National Institutes of Health (NIH) on giving an initial $1 million to BBN to use its time-sharing technology to develop a total hospital information system that would automate "the information gathering, processing, storage and retrieval that takes place in the modern hospital" [Baruch:1965]. The grand plan called for installing throughout the world-renowned Massachusetts General Hospital (MGH) Teletype terminals, which were to be connected to a central time-shared computer with a large mass-storage device (see figure 12.1). The first application areas to be automated included admissions/discharge, medication ordering and listing at the nursing station, and clinical chemistry laboratory test ordering and result reporting [Barnett:1967,Castleman:1969].

A team of about two dozen programmers, mostly in their early twenties, developed a complex set of inter-connecting application modules written in a macro assembly language. The high-morale spirit of the group– some of whom brought their dogs to work and many of whom worked until the morning's wee hours — had a "Children's Crusade" quality, a more innocent and less flashy version of the dot-com start-ups of the late 1990s. No one had much technical training: the original lead system-hardware/software guru, Shelly Boilen, was a former English major who then worked for an insurance company; his first superstar programmer, Bill Mann, had been a freshman MIT drop-out; and just two years out of Harvard College, I became project director.

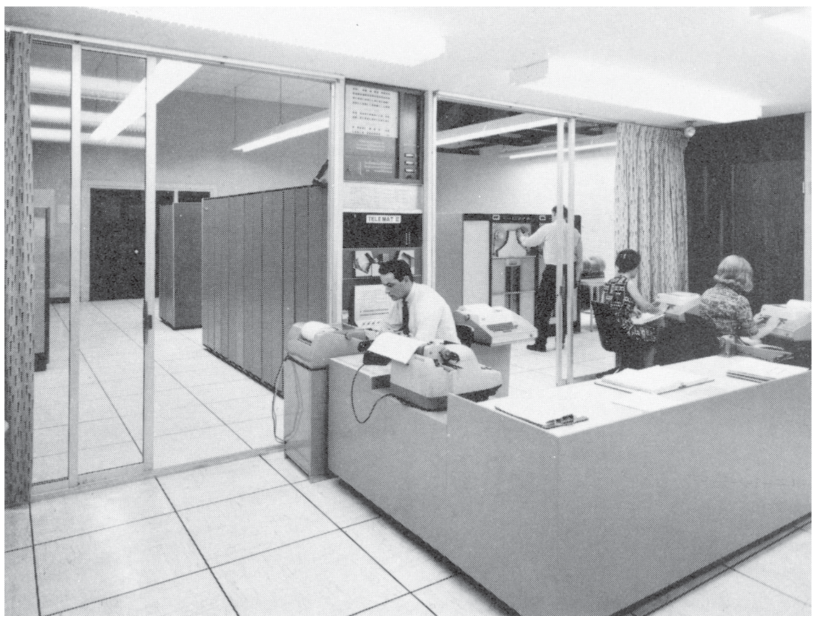


Figure 12.1. The Hospital Computer Project time-shared DEC PDP-1d located at BBN with a 50-megabyte specially built Fastrand drum for storing medical data files and 64 simultaneously usable telecommunication ports, many of which were connected to Teletype terminals operating at the MGH, 1966.

Some of the application areas, like medication handling, required considerable user input (entering all the prescriptions, recording each medication that was handed out) in order to generate for the nurse the listing of medication, patient, and times for distribution. This low ratio of output-to-input, plus the fact that it involved the busiest personnel (floor nurses) in a generally congested area (the nursing station), significantly reduced the perceived benefit of this application. In contrast, the lab reporting system was much more enthusiastically received and appreciated because the input was done by technicians in the relatively orderly central chemistry lab, while the output was simply printed at the nurse's station, creating the most legible and organized part of the patient's record, which could be easily scanned and digested by nurses, attending physicians, and medical students as they rotated through.

Using a teaching hospital as the first test site had its distinct advantages and disadvantages. While the intelligence, willingness to try new technologies, and general cachet of involving a prestigious medical icon worked to further the project, the complexity of the teaching hospital's organization and procedures, the strident internal politics and personalities, and the abundance of ego made it tough sledding. It is not surprising that the most successful commercial spin-off of this project (MediTech, Inc.), although founded by MGH staff who had worked on this project, consciously focused their marketing effort almost exclusively on *non*-teaching hospitals.

Approximately seven years after BBN started this Hospital Computer Project, the MGH, which had been building up a Laboratory of Computer Science under the energetic direction of Dr. G. Octo Barnett, took over the Project's operation and continuing development.

About three years later, in 1971, BBN began another initiative in computer application in patient care using time-shared remote-access computing — this time exploring ways to help the practicing office physician. The three-year government-sponsored project called CAPO (Computer Aids in the Physician's Office) installed several applications, principally an automated patient history-taking application in about a dozen private physician offices — both individual and group practices [Castleman:1974]. This patient-history application was designed to ease modification both of the text and branching structure of the patient on-line questionnaire and of the format of medical summary of the answers for the physician. While this modification capability was not extensively used, the apparent flexibility proved essential to user acceptance. Frequently, a prospective physician/user would look at CAPO's standard history protocol and say it was not at all something they could use; then after asking for what were often only a very few minor changes, the physician would be completely satisfied with their "custom" system. While the system was generally well received, without subsidy its cost was not low enough to justify for the average private physician, who is generally financially conservative.

For both CAPO and the Hospital Computer Project, the principal contribution toward computer-aided patient care was early exploration of feasible technologies, application approaches, and exposure for early-adopter users to evaluate. It is sobering to recall the optimistic predictions of the early days of computers in patient care — e.g., a total computerized patient record, completely integrated automation of all hospital medical processing, a national registry of all patient medical-records, all within ten or at most fifteen years — and to realize that even today (40 years later!) only fragments of these dreams exist.

12.2 Clinical research

Ironically, the most successful application of the Hospital Computer Project was not for patient care but for clinical research. The task of deriving useful trends, associations, overviews, and statistical summaries from sets of patient records is cumbersome for small sets, and practically impossible for large sets, of complex medical data without computer help. Virtually all teaching hospitals, medical schools, and pharmaceutical companies conduct extensive clinical research. By 1965 several clinical investigators at the MGH were using the Hospital Project's "Research System" [Allen:1966] to facilitate their research. The system permitted users to create data definitions and formats, to enter data whose syntactic validity could be verified, and then to retrieve subsets of records according to Boolean criteria. At the 1965 RAND/SDC conference on advanced data-management systems, BBN's Hospital Research System was the only system reported that operated interactively rather than in batch mode [Castleman:1965].

Two capabilities of this early system are especially noteworthy. One was the ability to specify new data fields as some mathematical combination of existing fields; this derived-data capability has continued to be a powerful feature in later clinical research systems, as well as in other software packages such as spreadsheet programs. The second important capability was the addition of a procedural application language to specify more complex specialized data manipulations and retrievals. BBN first adapted RAND's JOSS line interpreter [Shaw:1964] to the PDP-1 (calling the language TELCOMP) and then extended TELCOMP to handle text strings (STRCOMP) and organize medical data fields and files for specialized clinical research investigations (ISRCOMP and FILECOMP).[1] Neal Pappalardo, a software engineer at the MGH, under the direction of Drs. Octo Barnett and Jerome Grossman, led an effort to redesign, rewrite, and extend these language processors, which came to be known as MUMPS [Greenes:1969]. MUMPS was adopted by DEC as well as several computer medical service vendors, and became a widely used language for medical application computing.

In 1978, BBN began a decade-long project to develop, install, and support CLINFO systems [Gottlieb:1979] to aid clinical investigators at over 40 General Clinical Research Centers (GCRC's). GCRC's were special in-patient units in most of the major U.S. teaching hospitals with dedicated nurses, labs, and statisticians. Much of the nation's in-patient clinical research was conducted in GCRC's. CLINFO successfully helped not only with the analysis of clinical research data but also with many of the operational data-collection functions within the GCRC unit [Bilofsky:1980].

The effort was sponsored by the NIH, which employed a particularly effective procurement process. The external features of CLINFO had been specified by the RAND Corp under an earlier contract. The NIH then gave small short-term development contracts to two firms, each of which was to develop a demonstrable operable system by a deadline date. Then the sponsor held a "fly-off" (modeled after the DoD procurement of new aircraft where two or more competing manufacturers each build a prototype airplane to government specifications and then the one who does best in a fly-off competition wins the larger contract to build many more for operational use). In the CLINFO case, the winner would be funded to provide and support operational CLINFO systems at 40 sites around the country. While it was unusual for a non-DoD government agency to pay for multiple development efforts, in this case the NIH was able to extract prodigious productivity from the competitors. The literally round-the-clock drive to build a system to meet the specs by the deadline created an environment that was perhaps two-to-three times more productive than even the most well-done other government-funded development efforts. The specificity of the competition (detailed specs and firm deadline) and the single winner's prize (large long-term deployment contract) motivated the BBN medical software team, led by Chan Russell and David Fram, to achieve an astonishing level of productivity unmatched in the group's thirty-year history.

As discussed in Chapter 8, BBN research psychologists Drs. John Swets, Ron Pickett, and Dave Getty used computer technology in their investigations of medical diagnosis, imaging, and decision-making. One very interesting project showed that the computer-aided merged result of the independent x-ray readings by several general radiologists was as accurate as the reading of a single highly-skilled radiological specialist in the areas of mammography and prostate MRI's [Getty:1988].

12.3 Biomedical research

Throughout its history, BBN's senior corporate management maintained a strong interest in computer medical applications. In 1961 vice president Dr. Baruch initiated and directed the effort (until his departure in 1966 to start up a BBN commercial medical-computer joint venture with General Electric called Medinet). Then Frank Heart, a senior computer technologist/engineer at MIT's Lincoln Laboratory, joined BBN with a particular interest in developing computer technology for "life sciences." Even

[1]This sequence of programming language evolutions is detailed in Chapter 4.

though Frank Heart's major focus during his BBN tenure ended up being on developing packet-switched communication networks, he was also the corporate vice president responsible for the medical computer activities and made considerable contribution to these efforts.

One of Heart's contributions was to encourage the group to broaden beyond clinical data handling into more scientific areas of biomedical research. (This orientation is reflected in Heart's use of the term "life sciences.") For example, Frank was instrumental in expanding our activities into the then-infant field of genomics including our involvement in the first national genetic-sequence databank. In addition, Frank's uncanny ability to see which directions technology was heading, combined with an engineer's philosophy of build-it-simple, build-it-reliable, and build-it-usable, all contributed significantly to the effectiveness of BBN's medical computer work, especially in the application area of biomedical research. And finally, Heart was skilled in securing government R&D work and in interacting with the technical project officers; for example, his rapport with Dr. Bruce Waxman (one of NIH's foremost innovators and funders of medical computer R&D) was critical to BBN gaining participation in several projects.

In 1968 David Walden and I began consulting to Dr. William Raub, who worked for Bruce Waxman. (At the time Walden was a young programmer and Raub a junior NIH science administrator; it's interesting to note that Walden went on to manage major portions of BBN's business and Raub was for a time acting Director of National Institutes of Health.) Dr. Raub had a vision of using computer technology to help research pharmacologists better understand the relationships between the structure of small molecules (potential drugs) and their biological activity (what happens to you when you take some). He hoped that such structure/activity studies could lead to developing more effective drugs with fewer ill effects. Dr. Raub was also attracted to using computer-generated 3-D graphics to help investigators see the actual shape of the molecules in space (see figure 12.2), since the spatial orientation of the atoms in a drug molecule is often the most important determinant of a molecule's biological effects in the body.

After helping Dr. Raub specify an initial system, BBN went on to build and support a major biomedical resource called "PROPHET" [Castleman:1975]. Dr. Raub suggested the name; while not an acronym, the "PH" suggested pharmacology and the word "prophet" suggested advance into the future as well as a biblical allusion of helping lead researchers out of the wilderness. (And then it fell to BBN staff to remind every listener that the project was not about a commercial company's craving for the homonymous "profit".) For over 30 years this resource served thousands of biomedical investigators at more than one hundred medical research institutions. (In fact, this project may well be the longest major computer R&D contract the NIH has ever funded.)

The PROPHET Project produced several particularly notable innovations. First, as a result of his early consultations, Dave Walden proposed an extensible language (later when implemented by Fred Webb, it was named Parsec — see Chapter 21), which among other features supported special data types (e.g., a molecule data-type). Parsec in turn became the foundation for PL/ PROPHET [Castleman:1972] a richly featured high-level programming language that permitted rapid generation and modification of PROPHET application modules and eventually supported end-user programming of customized applications.

Then, in what may be the most far-reaching contribution BBN made in its thirty years of medical computer work, Howard Briscoe, a senior scientist/programmer, after visiting several potential PROPHET user sites, saw that most investigators kept their lab notebook data in tabular format (one row for each observation, one column for each type of data observed). PROPHET's resulting column-and-row table format (including

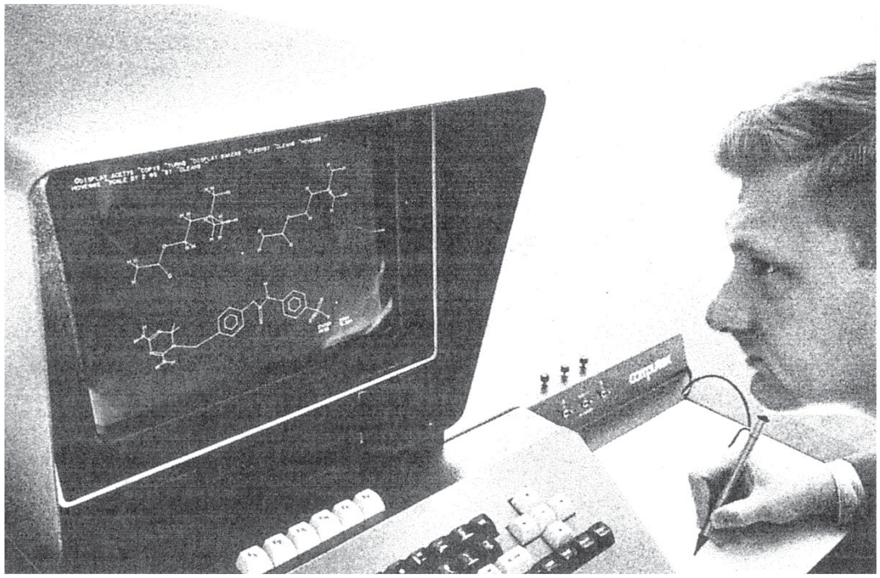


Figure 12.2. In the lower half of the terminal screen is a sketch of a molecular structure, which was entered into PROPHET using the tablet and stylus shown. In the upper half is a Dreiding model of acetylcholine generated by PROPHET from another sketch and displayed for stereoscopic viewing, 1972.

such features as derived columns) predated the first spreadsheet programs and went on to be the basis for BBN's commercially successful RS/1 software product.

BBN focused considerable energy on supporting the PROPHET users. Drs. Charlotte Hollister and James Wood operated a major user-support activity, which sent a BBN application scientist to visit virtually every PROPHET user site, sometimes as often as monthly. These visits were intended primarily to discover the directions in which the users wanted the system to evolve and secondarily to provide on-site user support. Hollister and Wood also organized an annual three-day user colloquium with both scientific and computer-technology related presentations, demonstrations, tutorials, and poster sessions. In some cases a BBN application scientist would collaborate directly with a PROPHET user; for example, BBN's Dr. Howard Bilofsky worked closely with Dr. Elvin A. Kabat, an eminent immunologist at Columbia University [Tai:1975]. Their co-authored PROPHET-produced database of immunoglobulin protein sequences was a widely used standard reference text for many years.

In 1980 BBN became involved in the environmental aspects of the life sciences. A PROPHET-based project for collecting data in the field was developed for the National Institute of Environmental Health Science; at the same time, the passage of the Toxic Substances Control Act led the Council on Environmental Quality to seek better ways to make information available to public interest groups, local governments and the public at large. Under the direction of Dr. Charlotte Hollister, this 5-year project, called CSIN (Chemical Substances Information Network) [Hollister:1985] utilized BBN's work in chemical information handling and user interface. BBN scientists developed a PC-based front-end software package to mediate between non-expert users and the vast information resources about chemical substances available through the American Chemical Society's Chemical Abstract Service, the National Library of Medicine, and commercial providers. This approach was eventually adopted by the NLM, which provided the broad-based support and access to make the system available nationwide.

Another BBN activity in biomedical research was the computer-systems side of the genetic sequence data bank known as GenBank [Bilofsky:1988]. Starting in 1982 BBN worked with Dr. Walter Goad at Los Alamos National Labs, who had been compiling what was to become the international repository for all reported sequences of nucleic acids (DNA and RNA). During its first five years of operation BBN had a $3 million contract to support and operate the GenBank database and distribution facility. Initially based within PROPHET, GenBank began as an informal collaboration with scientists at Los Alamos Labs. Drs. Howard Bilofsky and Wayne Rindone extended the techniques for storing immunoglobulin sequences first developed with Professor Kabat. Additional tools were added for searching, cataloging and publishing the data. Eventually, the maintenance of this rapidly growing database was turned over to the NIH. GenBank grew from an initial two thousand sequences in 1983 to over nine million sequences (ten billion base pairs) by the year 2000. (In 1990 Dr. Bilofsky went on to the European Molecular Biology Laboratory where he assisted in the creation of the European Bioinformatics Institute in Cambridge, UK.) GenBank continues today as one of the principal sources of biological sequence data and is recognized as an early contributor to the world-wide genomics revolution and to the success of the Human Genome Project.

Meanwhile, technology was changing and by 1985 the PDP-10 based PROPHET system, while still meeting the computational needs of many hundreds of scientists, was beginning to be a dinosaur in operational terms. After a lengthy review of the alternatives by a group of biomedical and computer scientists, the NIH decided to commission a re-implementation of PROPHET on UNIX-based graphics workstations. Under the direction of Robert Wells the underlying base was rewritten in the C programming language, but the PL/PROPHET language, in which so many higher-level applications had been written, was retained. This re-implementation allowed the user base of PROPHET to double within the space of a year after its release.

12.4 Commercial software products

There had always been a strong orientation in BBN's medical computer group to help the technology it developed be as widely deployed as possible so that it could have the greatest impact. Rather than seeing ourselves as researchers creating new knowledge, we viewed our contribution as creating computer environments that would support outside researchers in carrying out their work to uncover new scientific knowledge. For most of the 1960's and 70's we operated principally under the sponsorship of the National Institutes of Health and other health-related government agencies. But in the late 1970's, as Moore's law finally started to bring the cost of computer power within direct reach of most scientific investigators (via, for example, DEC's PDP-11 and VAX minicomputers), the opportunity for more widely deploying our technology via commercial software packages was created.

And so an effort to package the most useful and most widely usable technology into viable commercial software products began. Then as the activity started to show commercial promise, a separate new division (and later a separate corporation) was established. The key technical management (Paul Castleman, Chan Russell, David Fram) founded this new commercial activity in the early 1980's, and by the time they moved on to start up another BBN subsidiary in 1986, the company, now named BBN Software Products Corporation, was on its way to being ranked among the 50 largest independent software vendors in 1987 worldwide revenues [Software:1988]. By this time, the activity had become much more sales oriented under the professional sales management of Ean Rankin, Steve Lipsey, and Bruce Rampe.

BBN Software Products had two principal products. One was a scientific statistical data-analysis and graphical display package called RS/1 (originally named "Mission" [Castleman:1977]). Technically, RS/1 was the PROPHET software functionality (without the molecule-handling piece) recoded to operate on the new PDP-11 (and later VAX) minicomputers. While many of the earlier RS/1 users were scientists in the medical community and pharmaceutical industries, the application to manufacturing product design and quality control eventually dominated the RS/1 market.

The other major commercial software product, Clintrial, was targeted specifically to the pharmaceutical industry's clinical data-management needs during clinical trials for new drugs prior to submission to the FDA for approval. A crucial step in the process of building the Clintrial product was our assembling a consortium of four major drug companies to define the new system's functionality. While managing a consortium of independent industry scientist/managers was a bit like herding cats, it was well worth the effort in that we defined and then built a system that eventually captured a remarkable 85% of the world-wide market for pharmaceutical clinical data-management software.

While virtually all commercial enterprises, as well as most business-management books, profess putting the customer first, listening to the customer, and generally being customer-oriented, it seems such dicta are more often heard than actually followed, especially among R&D developers in leading-edge technology companies. More than the technical innovations, it was the actual continuing contact with the customer – an emphasis originally found so important in the PROPHET project – that may have been the principal factor in BBN Software Products' early success.

The customer-consortium method of developing Clintrial described above is one example. Another is the development in 1984 of RS/Expert, a statistical advisory system (based on the then-popular computer-science paradigm of "expert systems") targeted at scientists and engineers interested in performing design-of-experiments and in carrying out common data analysis and data modeling tasks. (The statistical foundations for RS/Expert [DuMouchel:1990] were designed by an energetic consulting statistician from MIT, Dr. William DuMouchel; DuMouchel, currently affiliated with AT&T Research and Lincoln Technologies Inc., has gone on to achieve a high degree of eminence in his field.) Through the financial entrepreneurship of BBN's CEO, Steve Levy, we raised $3.2 million via a privately funded limited partnership to fund RS/Expert's development. With the proposed RS/Expert fully funded, and with the investors looking for a rapid development time-scale so as to optimize their financial internal rate of return, once the first check was received and the technical team assembled, it was tempting to begin software design and development immediately. Instead, we arranged to spend most of the next three months visiting customers and getting feedback. Even the programmers went out on customer visits so they could gain a feel for the work environment of eventual RS/Expert users. The software was then developed (see figure 12.3) and successfully marketed; in fact, the limited-partner investors received a 300% return in less than three years.

The key personnel that started BBN Software Products all practiced this strong customer bias. I used to frequently say at company meetings, as well as to individual staff, "90% of what you need to know and don't already know to do your job is *out there*— so go out and meet more customers." I would try to lead by example and always highlighted my customer visits. BBN corporate president, Mike LaVigna, would evaluate any major new product initiative by trying to understand in depth what the real benefit to the customer would be and also whether that benefit was perceived by the customer as a strongly felt need. The head of software development, Chan Russell, and his principal application designer, Dave Fram, both spent significant time with customers. And BBN Software Product's first vice president of marketing and sales, Steve Lipsey, was relentless in promoting customer focus throughout the company.

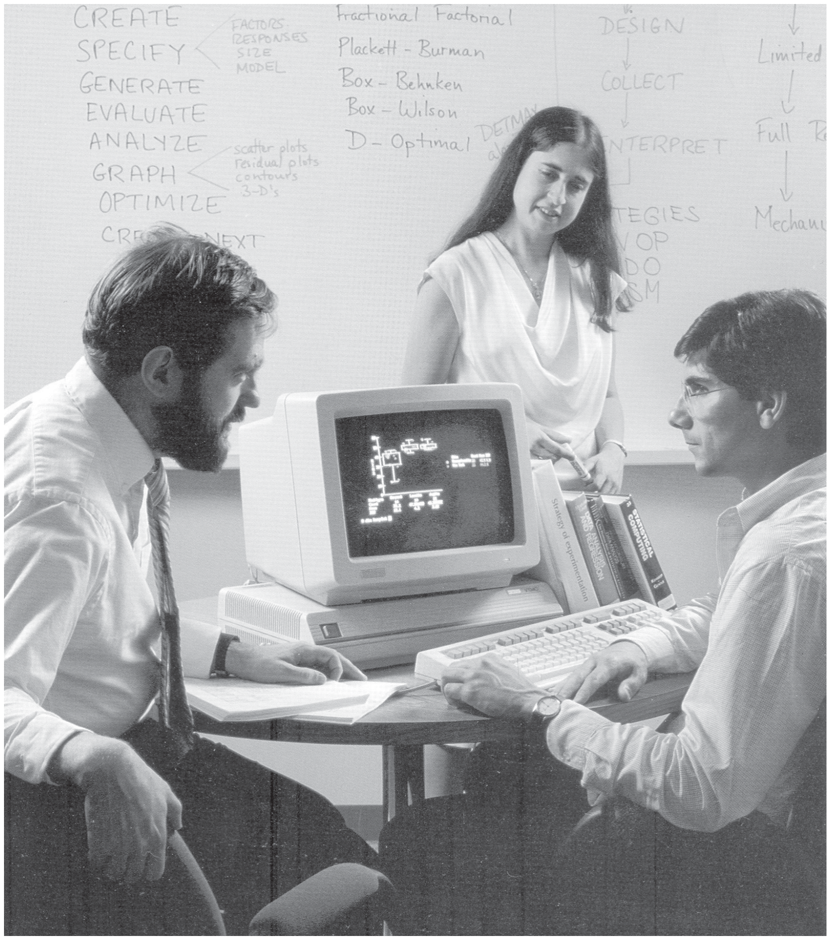


Figure 12.3. Senior software developers working on RS/Expert's menu-driven advisory system for guiding technical professionals through the statistical analysis of their data. The box plots on the screen help users to visualize the relationship between predictors (independent variables) and responses (dependent variables), 1986.

BBN has had impact on commercial ventures in the medical/computer area beyond its direct activities. In 1990 Chan Russell and I founded Belmont Research, Inc., which grew to become a leader in software and services for the drug industry. Two years after successfully selling Belmont Research in 1997, Russell, along with Dave Fram, started a second company called Lincoln Technologies, Inc.; later Dr. DuMouchel and I joined Lincoln's Board, and Lincoln has rapidly grown to become the principal provider of data-mining systems for drug safety to the FDA and to pharmaceutical companies.

12.5 Impact of the BBN environment

While the Hospital Computer Project was the largest single contract awarded to BBN during the 1960's, the medical computer activity was always a small part of BBN's total activities. As a result, the general BBN environment had a considerably larger impact on the medical computer group than the group had on BBN.

As with virtually all organizations, senior corporate management made a strong impact. Probably the largest influence was Frank Heart, who ran the corporate division in which the medical group operated for much of its existence. Heart, who was originally attracted to BBN principally because of its medical applications group, always took a strong interest in the group, even during those high-flying years that he led the ARPANET project. Heart prided himself on being an engineer — he called engineering the "art of the possible". (He would point out that while taking a military how-to-build-an-atom-bomb seminar he discovered it was "1% the magic of Einstein's physics, and 99% engineering".) His emphasis on developing solid workable systems ("not a toy", he would say), best exemplified by his success building the ARPANET, strongly and positively influenced the medical computer activities. For example, when someone might propose building a quick-and-dirty prototype, which could then be replaced by a real operational system at a later time, Heart would caution that, despite the best of intentions, his experience was that developers rarely end up completely starting over and redoing a system; more often, the original "prototype" just keeps getting improved. Heart frequently pushed for the development of systems that could be "widely deployed". (For someone with that as his primary goal, Frank Heart should be extraordinarily proud of his work on packet-switched networking.)

In my view, many of BBN's corporate management and division directors lacked a full appreciation for marketing (and its difference from sales); they were more orientated toward government-funded projects (vs. commercial customers); and they were attracted to the more intellectually exciting technologies and applications (vs. the more mundane ones that the commercial customer often wants and can actually use.) And while these inclinations were at times less helpful for BBN's medical computer activities, especially in its commercial initiatives, for most of the rest of BBN, they were actually a good thing; for example, there would have been no ARPANET/Internet without ignoring marketing justifications or commercial viability forecasts and without going way beyond the more tried and true technologies.

The principal figures at the top of BBN corporate management were chairman/CEO Steve Levy and president/COO Mike LaVigna. Both were strongly supportive of BBN's medical computer work despite its relatively modest size. Levy's particular genius was on the financial side. Long before the rise of the high-tech venture-capital industry, Levy used the little-known vehicle of private limited partnerships to fund promising technologies (e.g., RS/Expert) without compromising BBN's bottom line. His insight and timing in cash generation, which extended to successfully selling off ventures and raising considerable equity capital, kept BBN on firm, stable financial ground. Mike LaVigna was a born sales professional and helped bring to the function the needed appreciation, an attribute unfortunately often lacking in many high-tech R&D companies. Along with Steve Lipsey and Mike's protégé, Ean Rankin, LaVigna encouraged and facilitated the building of an impressive sales operation at BBN Software Products.

As with most corporate management, there were the occasional pushes for synergy among BBN's various groups (e.g., one group should use the technology of another); these were sometimes helpful, but more often annoying and diversionary. There was also little corporate emphasis on software usability and software development processes; while these can often morph into a bureaucratic nightmare, some enlightened attention might have been better than none. Finally, and most importantly, more marketing strength would have helped. We were never able to attract into management the marketing/business professionals of high enough caliber and position to impact BBN's bright pushy technical and financial management. This common problem is one factor why many excellent R&D labs have had difficulty carrying their ideas and inventions all the way through to widespread deployment and market acceptance (Xerox PARC being a prime example).

BBN had several faces. One face was as Cambridge's "Third University"; another was as a defense contractor/R&D lab. Each of these BBN faces, or actually cultures, impacted the medical computer activities both positively and negatively. Sometimes it felt like the medical group enjoyed the best of both worlds; at other times, it was more like steering between the Scylla and Charybdis of alien corporate cultures.

The academic culture at BBN certainly helped attract smart staff. Not only did the researchy image add a cachet in helping to hire exceptionally capable people, but it contributed to retaining them, thus keeping our staff turnover rate remarkably low. Just as Harvard, trading on its prestige, can hire technicians, programmers or teaching assistants for low wages, BBN managed to get the best people at reasonable salaries. But the medical group didn't want PhD computer scientists who just wanted a home to individually pursue original ideas of personal interest. Instead, we needed solid software developers and cross-disciplinary people willing and able to learn and interact with outside users in the medical community. We didn't want people who wanted to work principally on what they themselves found "interesting"; we didn't want people in love with their hyper-clever original idea or design (as is valued for an academic thesis). Instead, we needed people who understood that the best application ideas were "out there" among the potential users and not some far-out invention sprung fully formed from the head of a very smart computer techie. And again, unlike in the academic/thesis environment, the best software implementations were done in groups, producing well-documented easily understood code that was both reliable and extendable. Although we did not look for computer-science PhDs, we did seek out PhDs in disciplines related to our applications. For example, Drs. Howard Bilofsky and Charlotte Hollister each made very good use of their scientific training in chemistry in interacting with the research pharmacologists and in helping translate their needs into system specifications. While some of BBN's programs to support academic staff (the "Principal Scientist" position, the sabbaticals, the Science Development Program) occasionally created tension, resentments, and a feeling of inferiority in the non-academic side of BBN, these irritants were more often than not offset by the intellectual stimulation and general "classiness" of the Third University.

The fact that most of BBN operated as a defense contractor created a similar good-news/bad-news impact on the medical computing activities. There was a large reservoir of very talented technical professionals working on DoD contracts, many of whom were products of the 60's culture. Sometimes one or two of them would become available or decide that for ethical reasons they would rather work on medical rather than military applications. Another plus was the continuing flow of large, cost-plus defense contracts, which contributed to the company's overall financial health and stability. On the other hand, there was no question that BBN's principal client was DoD (for the computer side, DARPA in particular) and most of the corporate contacts and attention were (appropriately) focused there. Both DARPA project officers and the BBN academic computer scientists were strongly oriented toward exploring far-out technologies (looking for an order-of-magnitude leap); at times, the attractiveness and glamour of this push into the outer edges of technology spilled over into the medical computer group, causing us to be too far ahead of the usability curve. For example, the medical computer group's ventures into 3-D graphics — both software for rotating drug molecules and hardware to display true 3-D volumes [Sher:1988] for brain scans, molecules, and other applications — were way ahead of their time and were therefore unable to achieve wide user acceptance.

While BBN valiantly tried to shift its corporate culture to include commercial ventures, it had only mixed success. After leaving BBN in 1990, I did some public-interest work on this problem. At the time, "economic conversion of defense industries" was the hot topic (make trucks instead of tanks). I found that very few people in Washington DC had much direct experience with this issue and that they welcomed my sharing what I had learned at BBN. Through publishing newspaper op-eds [Castleman:1992a] and becoming an industrial policy adviser, first as staff in one presidential campaign and later informally to the White House, I tried to help policy-makers understand that successfully changing a corporate culture from government defense work to commercial activities is always very difficult, and often impossible. (As I was quoted in the St. Louis Post-Dispatch, “It’s not like changing your clothes; it’s more like changing your sex” [Castleman:1992b]). Even if the technologies are easily converted, the management/marketing/sales environment is sufficiently different so that it is often easier to start a new company than to try to convert an established company’s culture.

12.6 Observations and conclusions

From over 40 years involvement with medical application of computers I have, not surprisingly, assembled some generalizations on what seems to work and what doesn’t. My path has trod heavily in some areas of this sprawling field and only skimmed or skipped over many others, so these views are simply observations based on my individual professional experiences. The following are simple restatements of these views, without discussion, in the areas of applications, technologies, user communities, project activities, and staff.

Clinical research *applications* work well and can really help; often they involve adapting some established data-management technology to handle the quirks and idiosyncrasies of real-world clinical data. Hospital applications should be modular and start with the modules that are least invasive (e.g., labs, work flow, result reporting) even if they’re less flashy. Starting with building a total integrated system can be invasive and very difficult; for example, building a system to handle the whole patient medical record is a very big bite to start with. The benefits of any application module should be large and obvious (e.g., less hassles for the user, faster results, clearer reports, more insight), and the costs should be at most moderate (in learning time, change of work habits, monetary outlay, complexity of use). For example, esoteric graphs, fancy 3-D output, having to enter lots of data, all tend to be less well received.

The underlying *technologies* should be well established, reliable, and not mysterious. The technology should have been fully accepted in at least one other (non-medical) application area — that is, in successful routine operational use by people whose job is not trying out computer applications. Including an application-development procedural language (e.g., MUMPS, PL/PROPHET, and the RS/1 language called RPL) that application programmers (and sometimes sophisticated early-adopter users) can use to tailor applications is always very helpful and often essential. Simple data structures (e.g., 2-D tables) work better than more complex ones (e.g., more general networks of data connections). Developing the system to operate on hardware/system software that will stay (or become) popular is important; PROPHET on the PDP-10, RS/1 on the VAX, Clintrial using Oracle were all fortuitous choices (while in hindsight these may seem like obvious choices, at the time there was at least one other equally popular alternative to each choice). One often hears of worry that some initiative, whether it be technological or marketing, might be too late and miss the “window of opportunity”, my experience is that we were frequently too early — that is, we often ran headlong into the window before it even opened.

The choice of initial *user community* for a given system is perhaps more important than generally appreciated. Working with simpler and less arrogant medical centers, like community hospitals, can have distinct advantages over large eminent teaching hospitals. However, when choosing a first test-bed, it is often the teaching hospital that can afford the extra specialized staff and, more importantly, can attract the seed research funding, as was true with BBN's Hospital Computer Project with the Massachusetts General Hospital. For R&D organizations like BBN, it is generally much easier to initiate programs (e.g., the Hospital Computer Project, GenBank) as an early proof-of-concept than it is to succeed in wide-scale deployment to a broader community. In general, it is easier to have medical researchers, rather than practicing physicians or operational staff, involved in the initial system, especially when introducing new, unfamiliar complex systems or processes (e.g., PROPHET's work with Dr. Elvin Kabat, Clinfo's use of General Clinical Research Centers). Finally, there is a great temptation for the developers of a system to believe its potential user community can be much wider than originally planned. For example, RS/1, although originally planned for scientists, was technically a general-purpose data-analysis tool, which seemingly would be equally useful to other user communities such as in business or banking. However, it is in fact very very difficult to cross the industry-culture line; potential customers can sense whether the system and the people involved are of their culture. We quickly realized that, for example, to the financial analyst, RS/1 "smelled" like a technical/science system and would have a great problem in being accepted in an alien culture. The myth that because a system is technically general purpose, it therefore can be used in many different communities can be a dangerous marketing trap.

While many of the *project activities* required to successfully develop a medical application system are pretty straightforward, we learned several do's and don'ts. First, the temptation to push new flashy technology (e.g., the latest graphics display) is far less effective than the less glamorous job of slogging through organizing and baby-sitting meetings of potential users (e.g., the Clintrial consortium). Then once a system is deployed, putting considerable resources into substantive user-group meetings is well worth the effort. And to paraphrase the real-estate mantra, "Visit, visit, visit." Organizationally, as we transitioned from government-funded work into commercial products, there was a strong temptation to keep the activities combined (like keeping the PROPHET Project together with the commercial RS/1 activity) to avoid splitting up personnel and to ease the financial strain through economies of scale. My experience is that maintaining the combination beyond an initial start-up period is generally a big mistake. The cultures are sufficiently different, and it is already very difficult to grow a commercial-product culture within an R&D company, so the more separation in people, in organization and even in geography, the better.

Finally, styles and decisions about *staffing* are, of course, critical. While it is certainly important to hire bright energetic staff (we found the productivity among software developers can sometimes vary by an order of magnitude), it is also important for the staff involved in designing the functionality of an application not to think they know best — that is, to find staff who, although very smart, want to listen carefully to the potential users. We found it best to hire cross-disciplinary professionals who are oriented toward helping others, rather than being front and center themselves. Leaving the science or medicine to outside professionals in medical institutions but having staff that can communicate well with these collaborators seemed to work best.

Acknowledgments

It not possible to acknowledge individually the several hundred people who worked on BBN's medical-computer efforts during the past 44 years. In some cases, I have named individual contributors in the text of this paper. Among those many other BBNers who should also be specially acknowledged for their exceptional contributions

are Jeremy Pool, Art Gottlieb, Larry Sher, Sue Whitehead, Tom Kush, Bill Blackwell, Jon Cole, Alex McKenzie, Walt Bilofsky, Gail Rubin Walker, Dave Barach, Jim Pope, Bob Chatham, Harold Perry, Andee Rubin, Len Hantman, Hank Baig, Bernie Cosell, Bob Morgan, Steve Weiss, Fran Lewitter, Margaret Reid-Miller, Larry Polimeno, John Barnaby, and Stan Zdonik. My apologies to those many others whom it is not possible to name here, including those whose principal contributions were after the period of this history and also those many physicians, scientists and staff from outside universities, medical institutions, and federal agencies with whom we collaborated.

Editor's note. While Paul Castleman mentions Chan Russell several times in this chapter, he doesn't fully convey their synergistic relationship from the time Chan Russell joined Castleman's group at BBN in 1969, having recently graduated from Harvard. Within a few years, Russell became Castleman's informal business partner (although Castleman was officially the senior manager) as they undertook the sequence of opportunities described in this chapter and others. After they both left BBN, they became actual business partners with Russell now the top man on the operational organization chart. Castleman says, "We have been 'corporate spouses' for 35 years with never a sharp word."

References

Baruch, Jordan J. "Hospital Automation via Computer Time-Sharing," *Computers and Biomedical Research*, Eds. Ralph W. Stacey and Bruce D. Waxman, (New York: Academic Press, 1965), p. 291.

Barnett, G. O. & Castleman, P. A. "A Time-Sharing Computer System for Patient-Care Activities," *Computers and Biomedical Research*, Vol. 1, No. 1, March 1967, pp. 41-51.

Castleman, P. A., "A Time-Sharing Computer System for Hospital Information Processing," *Annals of the New York Academy of Sciences*, Vol. 161, Art 2, 1969, pp. 707-713.

Castleman, P. A. & Whitehead, S. F., "Evaluation of an Automated Medical History in Office Practice," *Proceedings of the Fourth Annual Conference of the Society for Computer in Medicine*, 1974.

Allen, S. I., Barnett, G. O., & Castleman, P. A. "Use of a Time-Shared General-Purpose File-Handling System in Hospital Research," *Proceedings of the IEEE,* Vol. 54, No. 12, December 1966, pp. 1641-1648.

Castleman, P. A. & Teitelbaum, S. G. "On-Line Data Management System for the Massachusetts General Hospital," *Proceedings of the Second Symposium Computer-Centered Data Base Systems*, Systems Development Corp., Santa Monica California, December 1965, Vol. 1, pp. 143-183.

Shaw, J. C. "JOSS: A design view of an experimental on-line computer system," *Proceedings of the Fall Jouint Computer Conference*, Vol. 26, part I, AFIPS Press, 1964, pp. 455-464.

Greenes, R. A., Pappalardo, A. N., Marble C. W., & Barnett, G. O. "A System for Clinical Data Management," *Proceedings of the Fall Joint Computer Conference*, AFIPS Press, 1969, pp. 297-305..

Gottlieb A., Fram D. et al. "CLINFO: A friendly computer system for clinical research," *Proceedings 12th Intl. Conf. on Medical and Biomedical Engineering*, Jerusalem, Israel, 1968.

Bilofsky, H. S. & Whitehead S. F. "CLINFO, A Clinical Research Data Management and Analysis System," *Proc. Fourth Annual Symposium Computer Applications in Medical Care Conf*, 1980, pp. 1286-1291.

Getty D. J., Pickett, R. M., D'Orsi, C. J., & Swets J. A. (1988). Enhanced interpretation of diagnostic images. *Investigative Radiology*, 23, 240-252.

Castleman, P.A.; Webb, F.N.; Hollister, C.; Siegel, J.R.; Zdonik, S.R.; and Fram, " D. M. The implementation of the PROPHET System" in *AFIPS - Conference Proceedings* 43, 457, 1974.

Castleman PA, Russell CH et al. (1972) An integrated command and procedural language for drug research. Proceedings of the ACM Annual Conference, pp. 202-223.

Wu, Tai Te; Kabat, Elvin A.; and Bilofsky, Howard "Similarities Among Hypervariable Segments of Immunoglobulin Chains", *Proceedings of the National Academy of Sciences of the United States of America*, 72, 5107-5110 (1975).

Hollister, C., Page-Castell, J. A. "The Chemical Substances Information Network: User Services Office. Evaluation and Feedback," *J. Chem. Info. Comp. And Sci.* Vol. 24, 1985, p. 259.

Bilofsky, H. S. and Burks, C. "The GenBank® genetic sequence data bank," *Nucleic Acids Research* Vol. 16, No. 5, 1988, p. 1861.

Software Magazine May 1988 cover story Vol 8, No 7. pp 61- 64.

Castleman P., Fram, D. et al. "MISSION: a low cost data management system for the biomedical community," *Proceedings of the First Annual Symposium on Computer Applications in Medical Care*, 1977, pp. 224-231.

DuMouchel, W. "The structure, design principles, and strategies of MULREG," *Annals of Mathematics and Artificial Intelligence, 2*: 1990, pp. 117-134.

Sher, L. "Spacegraph-a true 3D PC peripheral," in W. Robbins (ed.), *Proc. SPIE: Three Dimensional Imaging* Vol. 902, 1988, pp. 10-16.

Castleman, P. A. "Softening the Blow," *The Christian Science Monitor*, February 11, 1992, p. 18.

Castleman, P. A. "A Strategic Manufacturing Initiative," *St. Louis Post-Dispatch*, February 3, 1992.

Chapter 13

Educational Technology at BBN

Wallace Feurzeig

Since the early 1960s, BBN mathematicians, scientists, and engineers have explored ways to use computer and communications technologies to improve teaching and learning. BBN staff members have conducted basic research in human cognition and learning, developed innovative software tools to extend and enrich the traditional curriculum, and provided professional development to help educators make effective use of the new technologies. They have helped schools employ networking facilities to connect teachers and students with each other and with national and global resources.[1]

13.1 From drill-and-practice to "intelligent" CAI

No one, in the early sixties, saw the enormous potential of computers for education more clearly than J. C. R. Licklider. At that time, the prevailing model for the use of technology in education was drill-and-practice. The computer, like earlier electromechanical teaching devices, directed the interaction; it posed questions (typically multiple choice) and assessed the correctness of the student's answers. The student's role was purely responsive.

Paired-associate drill

An early example of this kind of computer-aided instruction (CAI) was the *paired-associate drill tutor* developed by Licklider in 1960 (Coulson, 1962). the program was used to provide practice in learning German language vocabulary. However, it could be used to provide practice in learning any kind of paired-associate material, that is, material that is organized in pairs, the first item of which is treated as a question, the second as an answer. The program worked roughly as follows. On each trial, the program would type the first item of a pair (i.e., the question) and wait for the student to type an answer. If the student typed the correct answer (i.e., the second item of the pair), the program would indicate that the response was correct, and move on to the next question. If the student gave an incorrect answer, the student was allowed to try again. If the answer was still incorrect, the program gave the correct answer, and proceeded to the next question. In commenting on the program's tendency to hold the attention of its users, Licklider made the following observation. "It seems possible, by exploiting the computer's constant capability for quick response and reinforcement, to develop techniques of instruction and intellectual exploration that will 'trap' the attention of students, and divert their energies from less constructive pastimes, to education."

[1] Editor's note: Color is important to many of the illustrations in this chapter. These may be seen at www.walden-family.com/bbn/feurzeig.pdf

Exploratory learning with graphs

Licklider was interested early in using computers for expressing multiple linked modes of representation of concepts and phenomena, especially including visual representations. In 1961 he developed a program, *Exploratory Learning with Graphs*, that enabled a user to specify a polynomial equation, such as the quadratic $y = a(x - b)^2 + c$, and assign values to the coefficients. The computer would generate the graph of the function. The user could then change the coefficients and the computer would generate the corresponding graph on the same screen. The intent was that, by exploring the effect on the graph of changes in the coefficients, and by investigating the operation of the program for a variety of polynomial functions, a student would develop a better intuitive understanding of the relationship between symbolic and graphic representations of functions.

Socratic System and the Mentor Language

Computer scientist Wallace Feurzeig came to BBN in 1962 to work with Licklider on the development of interactive computation facilities ("thin-skinned" computing) and user-oriented programming languages. After initial work programming acceptance routines for the newly arrived research computer, the Digital Equipment Corporation PDP-1, Feurzeig was invited by psychologist John Swets to collaborate on a CAI research project. The proposal called for the development of a conventional CAI system, very much like the drill-and-practice program described above. Feurzeig and Swets proposed an alternate approach. They wanted to extend the versatility and instructional power of then-current CAI systems by enabling computer support for complex learning tasks that allow students greatly enhanced capabilities for exploration and investigation.

In 1963, Feurzeig designed and implemented a CAI system with the following capabilities. Students would not be limited to responding to questions asked by the program. They could take the initiative by *asking* the questions — that would not be the sole prerogative of the program. This sharing of control between the program and the student was subsequently dubbed "mixed-initiative interaction." Further, the program would not have to make a fixed response to the student's inputs. Its response could be conditional on history (i.e., what had happened during the interaction thus far and thus, presumably, what the student had learned) as well as on the context within which the inputs occurred.

Swets named the program the *Socratic System* because of its ability to support sustained investigative dialogues between the student and the program. In a typical application, the program presents a problem to a student and engages him in a mixed-initiative dialogue in support of his attempt to solve the problem. The initial applications, designed to test the operation of the system, included an alphabet character recognition game and an electronic troubleshooting problem with a simple circuit. The major application, which demonstrated the power and potential usefulness of the system, was a differential diagnosis problem in clinical medicine. The application was inspired by a thought piece of Swets titled "Some Possible Uses of a Small Computer as a Teaching Machine." Here is an excerpt from that 1959 BBN memorandum.

> Let's say we want to make good diagnosticians out of our blossoming M.D.'s. So we have lots of cases in a computer, A student comes into the computer room, selects a card out of a file, and learns that John Doe has a medical history of thus and so, that some intern has "worked him up" on his recent admittance thus and so. What's John's problem? The student sits down at an available typewriter, and decides what else he wants to know. He wants to know if John has urea in his urine, so he asks the computer and the computer tells him the answer is "yes." "Aha, then how many

> white corpuscles does he have?" Answer: "150." "Well," he tells the computer, "this is clearly a case of mononucleosis." The computer replies: "Don't you think you ought to know whether John shows a Bobinski before deciding such?" "Yeah," says the student, "I guess so, does he?" Answer: "Yes." "Now I'm sure it's mononucleosis." "But" says the computer, "you are forgetting that John's pulse is normal, which you well know, is inconsistent with your diagnosis."...

In a *Socratic System* medical application (Feurzeig et al, 1964, Swets and Feurzeig, 1965), the student is given a statement of the problem (the patient's complaint and other presenting information) and a list of the questions and assertions that can be input by the student in the course of the interaction. Allowable questions include standard medical items: the patient's condition (e.g., general appearance?), physical examination (e.g., auscultation?), and requests for laboratory tests (e.g., rbc?). Allowable assertions include diagnoses (e.g., appendicitis), and justifications for a given diagnosis (e.g., evidence from urine culture). The list can be extensive. The student can do the history, physical exam, request lab reports, and make diagnoses in any order. As in real life, lab results need not be immediate, they may be reported back some time after they are requested. The program may respond differently to the same input, depending on occurrences in the interaction that should affect the student's knowledge. The problem can embody a situation that changes with time as the interaction develops. For example, the patient's condition may worsen or improve.

The system can respond to the student in the manner of a personal tutor, commending thoughtful questions, reproving unwarranted tests or foolish conclusions, acknowledging perceptive decisions, and questioning the grounds of diagnostic inferences. It can respond to a question by posing a question. The pedagogic strategies are not built in, they are specified by the instructor who designs the problem. Directed by these strategies, the system can develop interesting contingencies to considerable depth. The following excerpt from a student interaction with the system, illustrates some of these capabilities. The student's inputs are indented and capitalized to set them off from those of the system. The student's preliminary questions have been deleted.

Here is the problem statement.

It is mid-December. Late this afternoon a patient was admitted to the hospital. He is a well-developed, well-nourished, 34-year-old coal miner. He was brought in by his wife who reports that he has had "the grippe" for about 4 days. During the past 3 days she has given him fluids and aspirin every 6 hours with no sign of improvement. Today he suddenly became much worse. For the past 2 days he had complained of feeling very warm and of pain in his belly. This morning he had a severe shaking chill that was so bad his teeth rattled.

[*Deleted: student's initial requests for vital signs and other physical exam items.*]

Before proceeding with the reexamination, it might be instructive to pause and consider the diagnostic implications of the information at hand. Perhaps you think it too early to arrive at a definitive conclusion. On the other hand, you may be prepared to make the diagnosis now. Are you?

    NO.

Fine. Nevertheless, we've got a fair amount of information-at least enough to raise some, perhaps several, diagnostic possibilities. Have you any in mind?

    YES.

Name one.

    PNEUMOCOCCAL PNEUMONIA.

Good. The general signs of a pneumonic process are present. Of course, you haven't isolated the bug. Do you think the belly pain and flank pain reinforce the case for pneumococcal pneumonia?

NO.
Right. Though the white count is consistent, isn't it?
YES.
I think so too, but a later reading is indicated to confirm this finding. We've considered one possibility. Do you have another diagnosis in mind?
NO.
In that case, I'd like to talk about pulmonary infarction. What I like about that diagnosis is the presence of high fever, rusty sputum, and shaking chills. I don't like the dyspnea, pleural pain, and signs of consolidation-do you agree?
NO.
You're quite correct. I was just trying to see whether you were alert-and you were. Now let's see how the patient is today. Begin the physical when you're ready.
[*Deleted: student requests for vital signs and other physical exam items.*]
PNEUMOCOCCAL PNEUMONIA.
Yes. I remember you liked this diagnosis before. Are you absolutely sure?
NO.
What would help you nail down the diagnosis.
EVIDENCE FROM SPUTUM SMEAR AND CULTURE.
Good. Anything else?
EVIDENCE FROM X-RAY FINDINGS.
Good. X-ray shows area of consolidation at right lower lobe, consistent with a pneumonic process. Lab reports gram positive organisms in the sputum smear and the presence of pneumococci in the culture. Had you thought to rule out things like appendicitis and acute pyelonephritis? Apparently you weren't thrown off by the referred abdominal pain. In any case, you've made the correct diagnosis.

In this example the student was fairly insightful. Less-thoughtful students may make ill-informed diagnostic guesses. The program is more demanding when the evidence for their diagnoses is absent or weak.

Feurzeig designed and implemented a user-oriented programming language, *Mentor*, for developing *Socratic System* applications. One application was a parody of the Agatha Christie mystery, expressly designed to demonstrate the capabilities of both the *Socratic System* and the *Mentor* language. Other applications were made in diverse areas including classroom scheduling, management decision-making, and electronics troubleshooting. The work in medical diagnosis spearheaded further work in medical applications within BBN (e.g., the NIH hospital time-sharing project). It also led to extensive work on computers in medical education elsewhere in the 1970s and 1980s (Clancey, 1982).

Scholar program

Scholar was the first attempt to use a semantic network for knowledge representation as the basis of a teaching program. BBN computer scientist Jaime Carbonell designed the program in 1970. He distinguished frame-oriented CAI systems, which require that the designer anticipate all questions and answers in advance, from information-structure-oriented CAI systems, which represent domain information as a knowledge network from which specific system responses can be constructed. He realized that it was useful to have separate representations of the domain knowledge to be taught and the teaching strategy that was to be employed. Thus, *Scholar* has two components: a knowledge base to incorporate domain information, and a program for directing the tutorial interaction with the student. He saw the development of Scholar as "a step in the direction of CAI programs that know what they are talking about, the same way human teachers do."

The program was implemented in LISP. Its knowledge base was organized as a semantic network (Quillian, 1968). The program's initial knowledge base contained geopolitical information about South America. For each country it stored such information as population, land area, capital, major cities, major rivers, major mountains, major religions. What could be taught by the approach was limited to a set of facts: "The area of Peru is approximately 400,000 square miles. The major cities in Brazil are Sao Paulo, Rio de Janeiro, Brasilia, ..." The natural-language semantics and syntax required for treating such information are highly constrained. An excerpt from a representative student dialogue with Scholar follows (Carbonell, 1970). The student's inputs are indented and enclosed by asterisks.

What is Chile?
    a country
That's fine.

Select an alternative from the list: 214,000; 800,300; 49,883; 63,690;
For the question: approximately what is the area of Chile?
    800,300
Your answer is incorrect. The correct answer is 214,000.

The population in Chile is approx 850,000. True or False?
    true
You are correct. (sic!)

    Tell me something about Peru.
Wait a minute, I'm checking my files.
Let's proceed now.
Peru is a country.
It is in South America.
The population is approx 7,000.000 people.
The capital is Lima.

    Please tell me more about Peru.
The area of Peru is approx 48,000 square miles.
The language is Spanish.

    How many square milea are there in Brazil?
The area is approx 3,600,000 square miles.

Approximately what is the area of Argentina?

    What is the principal religion in Uruguay?
The principal religion in Uruguay is Catholicism.

Now answer the question you didn't answer before.
    The area of Argentina is 1,100,000 square miles.
Approximately true, but the exact value is 1,200,000 .

The eastern bordering country of Peru is ...?
    Catholicism.
Wrong. Catholicism is not a country but a religion.
The correct answer is: Brazil.

    Is Asuncion hot?
Since Asuncion is part of Paraguay, yes.

    What is the most important difference between Peru and Paraguay?
The primary difference is: the population of Peru is larger by a factor of 7.8.

Just before Carbonell's untimely death, he and BBN educational psychologist Allan Collins sought to use the *Scholar* system to explore a number of issues in natural language semantics (Carbonell and Collins, 1973). They had begun to consider how to implement some human reasoning capabilities in *Scholar*, such as the ability to make deductive, negative, and inductive inferences. They also intended to incorporate teaching strategies like those used by human tutors. Collins and colleagues sought to identify such strategies by recording and analyzing dialogues of human teachers tutoring students on South American geography (Collins, Warnock, and Passafiume, 1974). They identified six tutorial strategy categories: topic selection, interweaving questions and presentations, questioning about basic concepts, reviewing, use of hints, and responses to errors. They looked for common strategies across teachers and identified several general principles. For example, tutors appeared to introduce new questions when they thought the student knew the answer to previous ones, and to present additional information otherwise. They then attempted to program these strategies into Scholar.

One version of *Scholar* incorporated the capability to generate and display annotated maps (Collins, Adams, and Pew, 1978). It enabled the display of the entire continent of South America or sections appropriate to particular questions. For example, it could access the relevant map section and highlight the locations of cities in response to student requests such as "blink Brasilia." *Scholar* work with the South American geography knowledge base was augmented by two additional *Scholar* applications. One dealt with the structure of the ARPA computer network, the precursor of the Internet; the other with a text-editing system called NLS. The latter work resulted in the development of a version of *Scholar* called *NLS-Scholar* (Grignetti et al, 1974). *Scholar* proved influential in fostering research and development in the area of artificial intelligence applications to education, both outside and within BBN.

Why program

Why is a continuation of the work on *Scholar*. It was developed by Collins and fellow BBN psychologist Al Stevens after Carbonell passed away (Collins and Stevens, 1980). The major advance was in changing the character of the instruction from purely factual exchanges to causal reasoning interactions, as exemplified in the domain of meteorology. Here is an excerpt from a *Why* dialog.

Do you think the Amazon jungle has heavy or light rainfall?
    Heavy rainfall

Why does the Amazon have heavy rainfall?
    Because it has mountains nearby

Do you think that any place with mountains has heavy rainfall?
    Yes

Southern California has mountains.
Why doesn't Southern California have heavy rainfall?

The interaction illustrates the application of explicit teaching rules for generating the questions. These fall in the category of predictions, particular cases, prior causes, insufficient causes, and general rules. For example, the first question above asks the student for a prediction about a particular case. The second question asks for prior causes. The third question asks the student about a general rule. The last question introduces a counter-example to the student's insufficient causal response, and asks for prior causes.

The program was able to detect obvious student misconceptions. It was not used for carrying on extended dialogues nor did it claim to diagnose students' underlying misunderstandings or erroneous models of weather processes. Its major advance was the introduction of a tutorial strategy that employs a systematic logical approach for formalizing the questioning methods.

How the West Was Won

From 1973 through 1980, computer scientists John Seely Brown, Richard Burton, and their colleagues in the BBN Intelligent CAI group did advanced instructional research and software design leading to the implementation of tutorial systems incorporating powerful artificial intelligence facilities.

In 1975 they developed a paradigm for tutorial systems with capabilities for providing automatic feedback and hints in a game environment (Brown and Burton, 1975; Burton and Brown, 1976). They demonstrated the paradigm by implementing a computer coaching system,*West*, based on the children's game "How the West Was Won," a variation of the classic game "Chutes and Ladders." *West* was designed to teach computational skills through game playing strategy. There are two opposing players, one of whom may be the computer. The objective of the game is to land exactly on the last town on the game-board map. On each turn, a player spins three spinners to produce three numbers which he then combines using two of the operations addition, subtraction, multiplication, or division, possibly with parentheses. The value of the arithmetic expression thus generated is the number of spaces he gets to move. (Negative values result in backward moves). There are towns and shortcuts along the way to the goal. The rules specify the effect of a move landing on a town (moving to the next town), landing on a shortcut (advancing to the end of the row), or landing on the same place as his opponent ("bumping" him back two towns).

The system uses a computer-based "expert" player. It tracks and evaluates a student's moves and constructs a "differential model" that compares the expert's performance with that of the student. Procedural specialists assess the conceptual constraints that might prevent the student's full utilization of the environment. These help the tutor decide whether and when to suggest better moves to the student. For example, the student may be unaware of the benefit of bumping his opponent, e.g., of evaluating whether it is more advantageous to send her opponent back m places or to get ahead of her by n places. This assumes, of course, that she knows desirable values for m and n, and also how to construct appropriate arithmetic expressions that compute m and n from the three numbers selected by the spinners. Thus, a poor move might be due to the student's failure to consider a better alternative or to an incorrect computation of a move, a distinctly different kind of difficulty that calls for a qualitatively different instructional treatment.

Sophie

The intent of *West* was to turn a "fun" game into a productive learning environment without diminishing the student's enjoyment. The performance analysis in West identifies weaknesses in the student's play, but it does not diagnose the underlying difficulties that are responsible for them. From 1974 through 1978, the ICAI group undertook a considerably more ambitious effort, the development of an "intelligent" instructional system, *Sophie*, (for SOPHisticated Instructional Environment). Unlike previous CAI systems that employed AI methods to emulate a human teacher, *Sophie* sought to create a "reactive" environment that fosters a student's learning while he tries out his ideas working on a complex electronics troubleshooting task (Brown, Burton, and Bell, 1975). *Sophie* supports a student in a close collaborative relationship with an "expert" who helps the student explore, develop, and debug his own ideas.

Sophie incorporates a “strong” model of the electronics knowledge domain along with heuristic strategies for answering a student’s questions, critiquing his current solution paths, and generating alternative theories to his current hypotheses. Its expertise is derived from a powerful inferencing scheme that uses multiple representations of knowledge, including simulation models of its electronics circuit domain, procedural specialists for using these models, and semantic nets for encoding factual knowledge. *Sophie* was designed to demonstrate the feasibility of using AI techniques to construct an instructional system which, *on its own*, could reason, answer unanticipated questions, evaluate a student’s hypotheses, and critique the student’s performance behaviors, while carrying on an intelligent tutorial dialogue (Brown and Burton, 1978a).

In the basic scenario, *Sophie* acts as a lab instructor interacting with a trainee who attempts to debug a malfunctioning piece of equipment. The trainee can perform any sequence of measurements, ask questions about the implications of these measurements or more general hypothetical questions, and ask for advice about what to do next. *Sophie* may encourage the trainee to make a guess about what may be wrong given what he has found thus far. *Sophie* will evaluate his hypothesis by considering what he should have been able to derive from his current measurements, and if this information is contradicted by the hypothesis, *Sophie* will identify and explain these contradictions, just as in the Socratic System. *Sophie* can determine whether a new measurement is logically redundant and whether it rules out possible circuit faults which were still in contention. These capabilities require deep logical inferencing. They would be difficult or impossible to achieve by the use of pre-stored decision trees, since the sequence of measurements and the hypotheses any trainee will make is unknown in advance.

The electronic instrument used in *Sophie* is based on the Heathkit IP-28 power supply, a fairly sophisticated circuit with both current-limiting and voltage-limiting behavior. These two interrelated feedback loops make troubleshooting this circuit a nontrivial task. The following brief excerpt illustrates the kind of interaction made possible by *Sophie*. The student’s inputs are preceded by two carats. Reviewer annotations are parenthesized. The student has asked *Sophie* to insert a fault in the circuit. *Sophie* randomly selects a fault and inserts it into the model of the circuit.

The instrument has a fault and its present control settings are:

CC = 1.0	(Current Control - from 0.0 (Min Current) to 1.0)
CR = HIGH	(Current Range Switch, HIGH = 1 Amp, LOW = 0.1 Amp)
LOAD = 20	(Load Resistance in Ohms)
VC = 1	(Voltage Control - from 0.0 (Min Voltage) to 1.0)
VR = HIGH	(Voltage Range Switch, HIGH = 30, LOW = 10 Volts)

>>What is the output?
The output voltage is 11.7 volts.

>>What is it in a working instrument? (uses dialog model for interpreting pronouns)
In a working circuit the output voltage is 19.9 volts.

>>What is the voltage between R8 and Q1?
The voltage at N1 is 47.6 volts.

>>What is the base emitter voltage of the voltage limiting transistor? (=transistor)
The base emitter voltage of Q5 is .037 millivolts.

>>Is it possible that Q6 is open? (hypothesis assertion, questioned by *Sophie*)
Is the complete fault mode EB/EC/BC = OP/OK/OP what you intended?

>>Sure. (*Sophie* constructs a counter-example)
When the settings were: CC=1.0, CR=HIGH, LOAD=20, VC=1.0, VR=HIGH,
that fault conflicts with the following measurement. You observed that the

output voltage was 11.7 volts. If Q6 had fault mode EB/EC/BC = OP/OK/OP
it would be 23.9 volts. In a working circuit it is 19.9 volts.

(*Sophie* also determines what is "good" about the student's hypothesis.)

That fault does not explain any of the symptoms you observed but it is
consistent with the values for the base emitter voltage of Q5 and the voltage at N1.

Sophie employs a semantic grammar incorporating extensive domain-specific knowledge (Burton and Brown, 1979). Here are some representative examples of student inputs that are effectively and efficiently parsed by *Sophie*.

What is the voltage across the base emitter junction of the current limiting transistor?
What is the VBE of Q6?
What is current through the base of Q5?
What is the voltage between node 1 and the positive terminal of C6?
What is the dynamic resistance of R11?
What is the beta of the voltage limiting transistor?
In a working circuit what is the output voltage of the power reference transformer?
Change the output load to 10 megaohms.
Let C2 be leaky.
Set the current control to maximum.
Suppose the BE junction of Q6 is shorted.

Sophie has been used in a two-person gaming situation where one student introduces a fault into the circuit and predicts the consequences and the other student is challenged to discover the fault. The roles are then reversed. In another version of the game, one student introduces a circuit modification and the other requests measurements which the first student answers as best he can on the basis of his earlier prediction of the effects of his modification on circuit behavior. The system could monitor the operation and interrupt if a mistake could result in a serious compounding of misunderstandings.

The understanding capabilities in *Sophie* were largely based on its use of a general circuit simulation model (SPICE), together with a Lisp-based functional simulator incorporating circuit-dependent knowledge. These facilities were essential for inferring complex circuit interaction sequences such as fault propagation chains. *Sophie's* capabilities for modeling causal chains of events formed the basis for its explanation and question-answering facilities. *Sophie* used the simulator to make powerful deductive inferences about hypothetical, as well as real, circuit behavior. For example, it determined whether the behavior of the circuit was consistent with the assumption of specified faults and whether a student's troubleshooting inferences were warranted, i.e., whether the student had acquired information of the voltage and current states of relevant circuit components sufficient to unambiguously isolate the fault.

Sophie could infer what the student should have been able to conclude from his observations at any point, e.g. the currently plausible hypotheses and those that were untenable. However, because *Sophie* did not determine the reasons underlying the student's actions, e.g. the hypotheses he was actually considering, it was unable to diagnose the student's underlying conceptual difficulties in understanding and diagnosing circuit behavior. Despite this limitation, *Sophie* was one of the first instructional systems capable of supporting compelling and effective knowledge-based interactions, and it had enormous influence on other work in the ICAI area during the 1970s and 1980s.

13.2 Learning and teaching mathematics

Wallace Feurzeig founded the BBN Educational Technology Department (ETD) in 1965 to further the development of improvements in learning and teaching made possible by interactive computing and computer time-sharing. Time-sharing made feasible the economic use of remote distributed computer devices (terminals) and opened up the possibilities of interactive computer use in schools. The ETD work shifted from the development of tutorial environments to the investigation of programming languages as educational environments. The initial focus of the group was on making mathematics more accessible and interesting to beginning students.

Stringcomp

BBN programmers implemented the TELCOMP language in 1964 (Myer, 1966). It was modeled after JOSS, the first "conversational" (i.e., interactive) computer language, which had been developed in 1962-63 by Cliff Shaw of the Rand Corporation. TELCOMP was a FORTRAN-derived language for numerical computation. BBN made it available as a time-sharing service to the engineering market. Shortly after TELCOMP was introduced, Feurzeig extended the language by incorporating the capability for non-numerical operations with strings, to make it useful as an environment for teaching mathematics. The extended language was called *Stringcomp*.

In 1965–66, under U.S. Office of Education support, Feurzeig and his group explored the use of *Stringcomp* in eight elementary and middle school mathematics classrooms in the Boston area, via the BBN time-sharing system. Students were introduced to *Stringcomp*. They then worked on problems in arithmetic and algebra by writing *Stringcomp* programs. Experiencing mathematics as a constructive activity proved enjoyable and motivating to students, and the project strongly demonstrated that the use of interactive computation with a high-level interpretive language can be instructionally effective.

Logo educational programming environment

Feurzeig's collaborators in the development of *Logo* were BBN scientists Daniel Bobrow, Richard Grant, and Cynthia Solomon, and consultant Seymour Papert, who had recently arrived at MIT from the Piaget Institute in Geneva. The positive experience with *Stringcomp*, a derivative language originally designed for scientific and engineering computation, suggested the idea of creating a programming language expressly designed for children. Most existing languages were designed for doing computation rather than mathematics. Most lacked capabilities for non-numeric symbolic manipulation. Even their numerical facilities were typically inadequate in that they did not include arbitrary precision integers (big numbers are interesting to both mathematicians and children).

Existing languages were ill-suited for educational applications in other respects as well. Their programs lacked modularity and semantic transparency. They made extensive use of type declarations, which can stand in the way of children's need for expressing their ideas without distraction or delay. They had serious deficiencies in control structure, e.g. lack of support for recursion. Many languages lacked procedural constructs. Most had no facilities for dynamic definition and execution. Few had well-developed and articulate debugging, diagnostic, and editing facilities, essential for educational applications.

The need for a new language designed for, and dedicated to, education was evident. The basic requirements for the language were:

1. Third-graders with very little preparation should be able to use it for simple tasks
2. Its structure should embody mathematically important concepts with minimal interference from programming conventions
3. It should permit the expression of mathematically rich non-numerical algorithms, as well as numerical one

Remarkably, the best model for the new language (*Logo*) turned out to be *Lisp*, the lingua franca of artificial intelligence, which is often regarded by non-users as one of the most difficult languages. Although the syntax of *Logo* is more accessible than that of *Lisp*, *Logo* is essentially a dialect of *Lisp*. Thus, it is a powerfully expressive language as well as a readily accessible one.

The initial design of *Logo* came about through extensive discussions in 1966 among Feurzeig, Papert, and Bobrow. Papert developed the overall functional specifications, Bobrow did the first implementation (in *Lisp* on a Scientific Data Systems SDS 940 computer). Subsequently, Feurzeig and Grant made substantial additions and modifications to the design and implementation, assisted by Solomon and BBN engineers Frank Frazier and Paul Wexelblat. Feurzeig named the new language *Logo* ("from the Greek $\lambda o \gamma o \sigma$, the word or form which expresses a thought; also the thought itself," Webster-Merriam, 1923). The first version of *Logo* was piloted with fifth-and sixth-grade math students at the Hanscom Field School in Lincoln, Massachusetts in the summer of 1967, under support of the U.S. Office of Naval Research. (Feurzeig and Papert, 1968).

In 1967-68, the ETD group designed a new and greatly expanded version of *Logo*, which was implemented by BBN software engineer Charles R. Morgan on the DEC PDP-1 computer. BBN scientist Michael Levin, one of the original implementers of *Lisp*, contributed to the design. From September 1968 through November 1969, the National Science Foundation supported the first intensive program of experimental teaching of *Logo*-based mathematics in elementary and secondary classrooms. (Feurzeig et al, 1969). The seventh grade teaching materials were designed and taught by Papert and Solomon. The second grade teaching materials were designed by Feurzeig and BBN consulting teacher Marjorie Bloom. The teaching experiments demonstrated in principle that *Logo* can be used to provide a natural conceptual framework for the teaching of mathematics in an intellectually, psychologically, and pedagogically sound way.

Classroom work to investigate the feasibility of using *Logo* with children under ten years old was first carried out at the Emerson School in Newton, Massachusetts in 1969. The students were a group of eight second-and-third graders (ages seven to nine) of average mathematical ability. The children began their *Logo* work using procedures with which most children are familiar. Examples included translating English into Pig Latin, making and breaking secret codes (e.g., substitution ciphers), a variety of word games (finding words contained in words, writing words backwards), question-answering and guessing games (Twenty Questions, Buzz, etc.). Children already know and like many problems of this sort. Children think at first that they understand such problems perfectly because, with a little prodding, they can give a loose verbal description of their procedures. But they find it impossible to make these descriptions precise and general, partly for lack of formal habits of thought, and partly for lack of a suitably expressive language.

The process of transforming loose verbal descriptions into precise formal ones becomes possible and, in this context, seems natural and enjoyable to children. The value of using *Logo* becomes apparent when children attempt to make the computer perform their procedures. The solutions to their problems are to be built according to a preconceived, but modifiable plan, out of parts which might also be used in building other solutions to the same or other problems. A partial, or incorrect, solution is a useful object; it can be extended or fixed, and then incorporated into a large structure. Using procedures as building blocks for other procedures is standard and natural in *Logo* programming. The use of functionally separable and nameable procedures composed of functionally separable and nameable parts coupled with the use of recursion, makes the development of constructive formal methods meaningful and teachable.

The work of one of the seven-year-olds, Steven, illustrates this course of development. Steven, like all second-graders in the group, was familiar with the numerical countdown procedure accompanying a space launch. He had the idea of writing a COUNTDOWN program in Logo to have the same effect. His COUNTDOWN program had a variable starting point. For example, if one wished to start at 10, he would simply type COUNTDOWN 10, with the following result:

```
10 9 8 7 6 5 4 3 2 1 0 BLASTOFF!
```

He designed the program along the following lines. (The English paraphrase corresponds, line by line, to Logo instructions.)

```
TO COUNTDOWN a number
Type the number.
Test the number: Is it 0?
If it is, type "BLASTOFF!" and then stop.
If it is not 0, subtract 1 from the number, and call the result "newnumber".
Then do the procedure again, using :newnumber as the new input.
```

Steven's program, as written in Logo, followed this paraphrase very closely. (The colon preceding a number name designates its value. Thus, :NUMBER is 10 initially.)

```
TO COUNTDOWN :NUMBER
1 TYPE :NUMBER
2 TEST IS :NUMBER 0
3 IFTRUE TYPE "BLASTOFF!" STOP
4 MAKE "NEWNUMBER" DIFFERENCE OF :NUMBER AND 1
5 COUNTDOWN :NEWNUMBER END
```

(Note that the procedure calls itself within its own body — it employs recursion, however trivially.) Steven tried his procedure. He was pleased that it worked. He was then asked if he could modify COUNTDOWN so that it counted down by 3 each time, to produce

```
10 7 4 1 BLASTOFF!
```

He said "That's easy!" and he wrote the following program.

```
TO COUNTDOWN-3 :NUMBER
1 TYPE :NUMBER
2 TEST IS :NUMBER 0
3 IFTRUE TYPE "BLASTOFF!" STOP
4 MAKE "NEWNUMBER" DIFFERENCE OF :NUMBER AND 3
5 COUNTDOWN :NEWNUMBER
END
```

He tried it, with the following result,

```
COUNTDOWN-3 10
10 7 4 1 -2 -5 -8 -11 -14 -17 . . .
```

and the program had to be stopped manually. Steven was delighted! When he was asked if his program worked, he said "No." "Then why do you look so happy?" He replied "I heard about minus numbers, but up till now I didn't know that they really existed!"

Steven saw that his stopping rule in instruction line 2 had failed to stop the program. He found his bug — instead of testing the input to see if it was 0, he should have tested to see if it was negative. He changed the rule to test whether 0 is greater than the current number,

```
2 TEST IS :NUMBER LESS-OR-EQUAL 0
```

and then tried once more.

```
COUNTDOWN-3 10
10 7 4 1 BLASTOFF!
```

And now COUNTDOWN-3 worked.

Steven then worked on an "oscillate" procedure for counting up and down between two limits by a specified number of units. Two special and characteristic aspects of programming activity are shown in Steven's work — the clear operational distinction between the definition and the execution of a program, and the crucial mediating role served by the process of program "debugging."

Logo-controlled robotturtles were introduced in 1971, based on work of BBN and MIT consultant Mike Paterson. Screen turtles were introduced at MIT around 1972. BBN engineer Paul Wexelblat designed and built the first wireless turtle in 1972. He dubbed it "Irving." Irving was a remote-controlled turtle about one foot in diameter. It was capable of moving freely under *Logo* commands via a radio transceiver attached to a teletype terminal connected to a remote computer. Irving could be commanded to move forward or back a specified increment of distance, to turn to the right or left a specified increment of angle, to sound its horn, to use its pen to draw, and to sense whether contact sensors on its antennas have encountered an obstacle.

Children delighted in using Logo to command Irving to execute and draw patterns of various kinds. An early task started by having them move Irving from the center of a room to an adjoining room — this typically required a sequence of ten or fifteen move and turn commands. After Irving was somewhere out of view, the child's task was to bring Irving back home, to its starting point in the original room. This had to be done only through using *Logo*, without the child leaving the room or peeking around the doorway! The child had a complete record of the sequence of commands she had used since each command she had typed was listed on the teletype printer.

This task was fascinating and, for all but the most sophisticated children, quite difficult. The notion that there is an algorithm for accomplishing it was not at all obvious. They knew that Move Forward and Move Back are inverse operations, as are Right Turn and Left Turn. But they didn't know how to use this knowledge for reversing Irving's path. The algorithm — performing the inverse operations of the ones that moved Irving away, but performing them in the reverse order — is an example of a mathematical idea of considerable power and simplicity, and one that has an enormous range of application. The use of the turtle made it accessible to beginning students. Once the children were asked how to make Irving just undo its last move so as to get to where it had been the step before the last, most children had an "Aha" experience, and immediately saw how to complete the entire path reversal. The algorithm was easily formalized in *Logo*. This paved the way to understanding the algebra procedure for solving linear equations. The algorithm for solving a linear equation is to do the inverse of the operations that generated the equation, but in the reverse order — the same algorithm as for path reversal.

These examples illustrate the kinds of interactions that have been fostered through work with *Logo*. There is no such thing as a typical example. The variety of problems and projects that can be supported by *Logo* activities, at all levels of mathematical sophistication, is enormous. *Logo* has been the center of mathematics, computer science, and computational linguistics courses from elementary through undergraduate levels. (Feurzeig and Lukas, 1972; Abelson and DiSessa, 1983; Lukas and Lukas, 1986; Goldenberg and Feurzeig, 1987; Lewis, 1990; Cuoco, 1990; Harvey, 1997)

In 1970, Morgan and BBN software engineer Walter Weiner implemented subsequent versions of *Logo* on the DEC PDP-10 computer, a system widely used in universities and educational research centers. Throughout 1971-74, BBN made DEC 10 *Logo* available to over 100 universities and research centers who requested it for their own research and teaching. In 1970, Papert founded the Logo Laboratory at MIT, which further expanded the use of Logo in schools. The advent of micro-computers, with their wide availability and affordability, catapulted *Logo* into becoming one of the world's most widely used computer languages during the 1970s and 1980s and, especially in Europe, currently.

Algebra Workbench

Introductory algebra students have to confront two complex cognitive tasks in their formal work: problem-solving strategy (deciding what mathematical operations to perform in working toward a solution) and symbolic manipulation (performing these operations correctly). Because these two tasks — each very difficult in its own right for beginning students — are confounded, the difficulties of learning algebra problem solving are greatly exacerbated. To address these difficulties, Feurzeig and BBN programmer Karl Troeller developed the *Algebra Workbench.* The key idea was to facilitate the acquisition of problem-solving skills by sharply separating the two tasks and providing students automated facilities for each (Feurzeig and Richards, 1988).

The program includes powerful facilities for performing the symbolic manipulations requested by a student. For example, in an equation-solving task it can add, subtract, multiply, or divide both sides of the equation by a designated expression, expand a selected expression, collect terms in an expression, do arithmetic on the terms within an expression, and so on. This enables students to focus on the key *strategic* issue: choosing what operation to do next to advance progress toward a solution. The program will carry out the associated manipulations. The *Workbench* has a variety of facilities to support students' work. It can advise the student on what would be an effective action at any point; it can check a student's work for errors, either at any point along the way, or after the student completes his work; and it can demonstrate its own solution to a problem.

The *Algebra Workbench* was designed for use with formal problems in the introductory course, e.g., solving equations and inequalities, testing for equivalence of expressions, factoring, simplification, etc. It can provide a student with a set of problems, such as: $(n-1)/(n+1) = 1/2$, or $(16x+9)/7 = x+2$, or $10y-(5y+8) = 42$. It can accept other problems posed by the student or teacher. In demonstrating the working out of a problem, it employs pattern recognition and expression simplification methods at a level that can be readily emulated by beginning students. Its facilities for expression manipulation, demonstration, explanation, advice, and critical review are available at the student's option at any time during a problem interaction. See Figure 13.1.

Another student, who worked on the same problem, replaced $2*(n-1)$ by $2n-1$ on the left side of the equation, transforming it into $2n-1 = n+1$. When the student announced that he had solved the equation with the result $n = 2$, the system reviewed his work and pointed out his incorrect expansion of the left-side expression during the initial step.

A commercial version of the *Algebra Workbench* was developed by BBN scientist John Richards and Feurzeig under support of Jostens Learning Corporation in 1993.

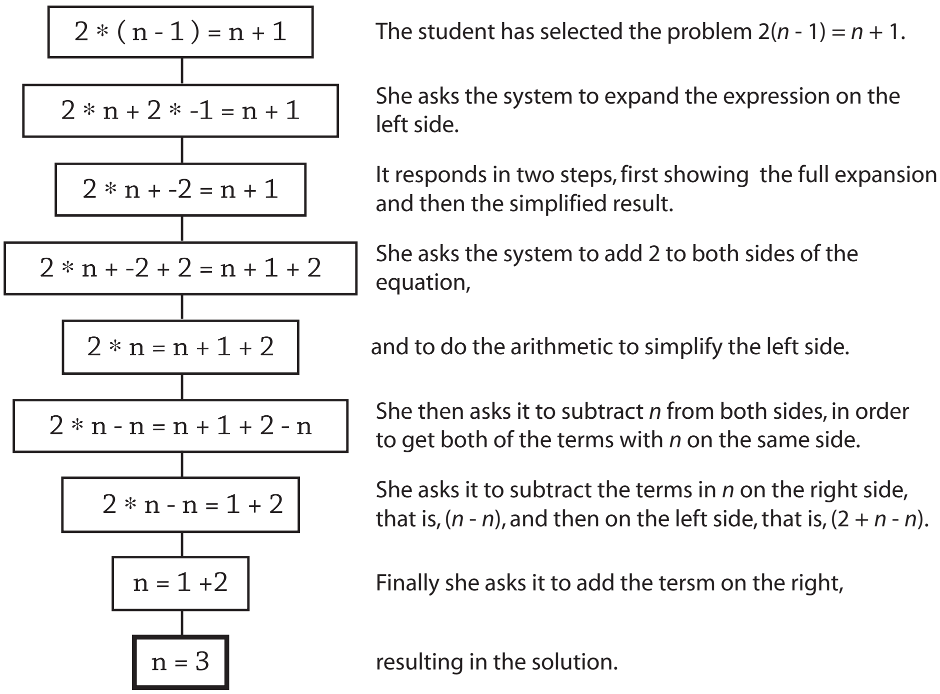


Figure 13.1 Transcript illustrating use of the *Algebra Workbench*.

The system was never released because Jostens was absorbed into a larger publishing firm that was not interested in educational software products.

Non-prescriptive learner-centered systems with instructional supports (like the *AlgebraWorkbench*) can help students acquire the knowledge and skills they need to use mathematics. The learning benefits of these tools are further enhanced when they are augmented by open-ended learner-driven tools like Logo that enable students to undertake projects with mathematically rich content in broader contexts.

Buggy and Debuggy

Buggy was a computer game based on the diagnostic interactions of a teacher and a simulated student. The teacher is presented with an arithmetic problem that the "student" has done incorrectly, due to an underlying bug in one of the arithmetic subprocedures. The teacher's task is to diagnose the bug by providing strategic test problems for the "student" to solve in order to discover the bug. The problems given by the teacher are answered by the "student" using the buggy procedure. When the teacher thinks he knows the answer, *Buggy* asks the teacher to describe what he thinks the bug is by responding to a five-problem test in the same way that the student would do them, i.e., by simulating the student's bug (Brown and Burton, 1978b).

The following is a protocol illustrating the use of *Buggy* by a team of teachers. The teachers' inputs are preceded by vertical lines on the left.

I HAVE CHOSEN A BUG. HERE IS AN EXAMPLE OF THE BUG.

```
 17
 +5
---
 13
```

NOW YOU CAN GIVE ME PROBLEMS TO DETERMINE THE BUG.

```
|   18     43
|   +6    +79
    ---    ---
|   15     23
```

| Got the bug!
HERE IS A PROBLEM TO TEST YOUR THEORY ABOUT THE BUG. WHAT IS:

```
|    21
|   +39
     ---
|    51
```

THAT'S NOT THE BUG I HAVE. USING MY BUG:

```
|    21
|   +39
     ---
|    15
```

TRY GIVING ME SOME MORE EXAMPLES.

```
|       51    99     68
|    +1707   +99     +9
       ---   ---    ---
|       21    36     23
```

| Got the bug!

PLEASE DESCRIBE THE BUG YOU THINK I HAVE.
| The student is adding all the digits together.

HERE ARE SOME PROBLEMS TO TEST YOUR THEORY. WHAT IS:

```
|    33   1091     8    28     90
|   +99    +60   +34   +70     +6
    ---    ---   ---   ---    ---
|    24     17    15    17     15
```

VERY GOOD. MY DESCRIPTION OF THE BUG IS: THE STUDENT
ALWAYS SUMS UP ALL THE DIGITS WITH NO REGARD TO COLUMNS.

The *Buggy* data base incorporated a substantial number of typical student bugs in addition and subtraction, based on empirical studies of the buggy behaviors of elementary students of arithmetic. The data base consisted of 20,000 problems performed by 1300 students (Brown et al, 1977).

The work on *Buggy* motivated the development of *Debuggy*, a diagnostic modeling system for automatically synthesizing a model of a student's bugs and misconceptions in basic arithmetic skills (Brown and Burton, 1978). The system introduced procedural networks as a general framework for representing the knowledge underlying a procedural skill. Its challenge was to find a network within this representation that identified the particular bugs in a student's work as well as the underlying misconceptions in the student's mental model.

Summit

In 1983, Feurzeig and BBN cognitive scientist Barbara White developed an articulate instructional system for teaching arithmetic procedures (Feurzeig and White, 1984). *Summit* employed computer- generated speech and animated graphics to aid elementary school children learn standard number representation, addition, and subtraction. The system comprised three programs. The first was an animated bin model to help students understand place notation and its relationship to standard addition and subtraction procedures. It could display up to four bins on the screen: a thousands bin, a hundreds bin, a tens bin, and a ones bin. The bin model in Figure 13.2 represents the number 2934.

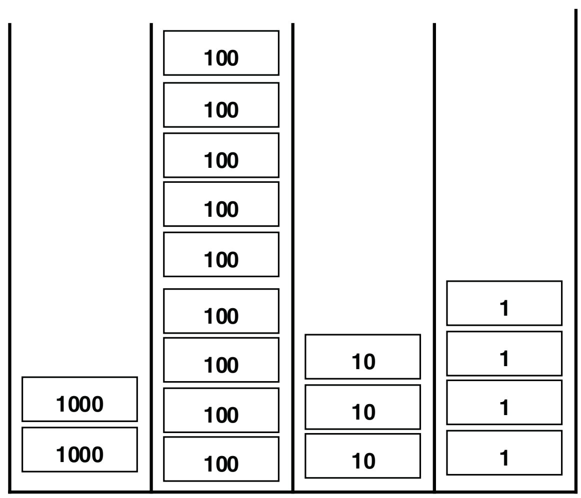


Figure 13.2 The *Summit* bin model.

The student could give commands such as ADD 297 or SUBTRACT 89 and the program would cause the appropriate numbers of icons to be added or subtracted from the appropriate bins graphically, in a manner analogous to the standard procedures for addition and subtraction. When a bin overflowed, the program animated the carrying process. Similarly, in subtraction when there were not enough icons in the appropriate bin, the model animated the process of "borrowing" (replacing). The program explained its operations to the student using computer-generated speech.

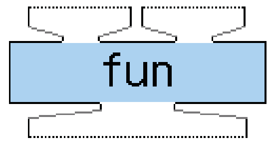
The second program in *Summit* demonstrated the standard (right-to-left) algorithms for addition and subtraction. It displayed a problem on the screen and talked its way through to the solution, explaining its steps along the way. The demonstrations were sequenced, starting with single-digit problems and working up to four-digit problems. They included explanations of the more difficult cases, such as borrowing across zeros.

The third program in *Summit* gave students an opportunity to practice addition and subtraction problems aided by feedback and guidance. A student, using a computerized work tablet, worked through a problem using keyboard arrows to control the positioning of number entries. If the student made an error, he was presented with the choice of trying it again or seeing a *Summit* demonstration of how to solve the problem.

Summit was an exploratory research project. Few elementary schools had computers, and virtually none had computer-generated speech facilities. The *Summit* programs were written in Logo on the Apple II computer. The system was tested with fourth-grade students in a Cambridge, Massachusetts school in 1983 and proved effective in helping students learn place notation and the standard algorithms for addition and subtraction.

Function Machines

Function Machines is a visual programming environment for supporting the learning and teaching of mathematics. It is a dynamic software realization of a function representation that dates back to the 1960s — the notation that expresses a function as a "machine" with inputs and outputs.



Function Machines employs two-dimensional representations — graphical icons — in contrast with the linear textual expressions used for representing mathematical structures in standard programming languages. The central *Function Machines* metaphor is that a function, algorithm, or model is conceptualized as a machine, displayed as shown in the figure, where fun represents a function. The two funnel-shaped objects at the top of the figure are called input hoppers; the inverse one at the bottom is called an output spout.

Machines can have multiple inputs and outputs. The output of a machine can be piped from its output spout into the input hopper of another machine. A machine's data and control outputs can be passed as inputs to other machines through explicitly drawn connecting paths. The system's primitive constructs include machines corresponding to the standard mathematical, graphics, list processing, logic, and I/O operations found in standard languages. These are used as building blocks to construct more complex machines in a modular fashion. Any collection of connected machines can be encapsulated under a single icon as a higher-order "composite" machine; machines (programs) of arbitrary complexity level can be constructed (Feurzeig, 1993, 1994a).

Execution is essentially parallel — many machines can run concurrently. The operation of recursion is made visually explicit by displaying a separate window for each instantiation of the procedure as it is created, and erasing it when it terminates. The hierarchical organization of programs implicit in the notion of a composite machine fosters modular design and helps to organize and structure the process of program development. Like a theater marquee, the system shows the passage of data objects into and out of machines, and illuminates the active data and control paths as machines are run. Thus, it visually animates the computational process and makes the program semantics transparent.

Function Machines supports students in a rich variety of mathematical investigations. Its visual representations significantly aid one's understanding of function, iteration, recursion, and other key computational concepts. It is especially valuable for developing mathematical models. To understand a model, students need to see the model's inner workings as it runs. At the same time they need to see the model's external behavior, the outputs generated by its operation. Function Machines supports both kinds of visualizations. The use of these dual-linked visualizations has valuable learning benefits.

Function Machines was designed in 1987 by members of the BBN Education Department including Feurzeig, Richards, scientists Paul Horwitz and Ricky Carter, and software developer Sterling Wight. Wight implemented *Function Machines* for Macintosh systems in 1988. A new version, designed by Feurzeig and Wight, was completed in 1998. It is implemented in C++ and runs both on Windows and Macintosh systems.

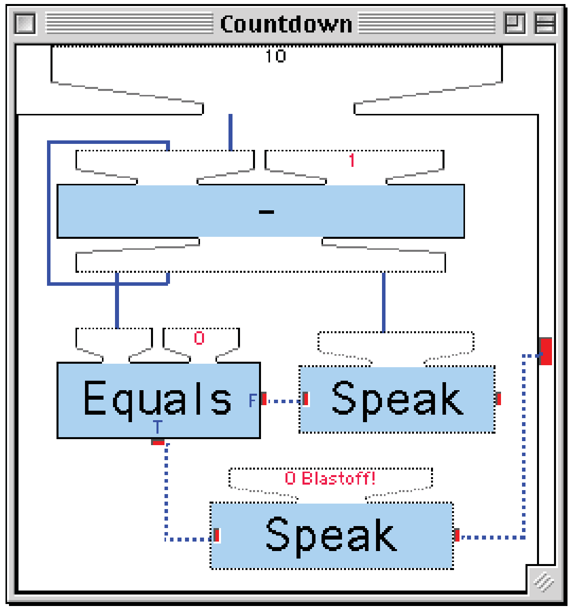


Figure 13.3 *Function Machines* Countdown program.

Figure 13.3 shows a Function Machines Countdown program, corresponding to the first of the Logo Countdown programs written by second grader Steven, which was illustrated on page 292.

The Countdown composite machine comprises four machines: a subtraction machine (−), an Equals machine, and two Speak machines. The Speak machines use speech generator software to "speak" their inputs. Countdown has as its input the value 10, which is sent to the left hopper of the −machine. The right hopper of that machine has as its input the constant 1 (the decrement of the subtraction operation). The output of the −machine is sent to three places: the left hopper of the Equals machine, the hopper of a Speak machine, and back to its own left hopper as its next input. When Countdown runs, it tests to see if the output of the −machine is equal to 0. If not, it "speaks" that output value; otherwise (if the output is 0), it speaks "0 Blastoff!" and triggers the stop button (in red, on the right border). During each iteration, the output of the −machine is decreased by 1, and the process is repeated, terminating when the output becomes 0.

Function Machines has been used in elementary and secondary mathematics classrooms in the United States, Italy, and Germany (Feurzeig, 1997, 1999; Cuoco 1993, Goldenberg, 1993). The following activities from a fifth-grade pre-algebra sequence (Feurzeig, 1997) illustrate the spirit and flavor of the approach. Students begin using the *Function Machines* program Predicting Results shown in Figure 13.4. Students enter a number in the Put In machine. It is sent to the +machine, which adds 2 to it; the result is then sent to the ∗machine, which multiplies it by 5; that result is sent to the Get Out machine.

The display window on the right shows a series of computations; thus, when students put in 5 the machine gets out 35. The program uses speech. When 5 is entered, the program gives the spoken response Put In 5 and, as the result of that computation is printed, the program responds Get Out 35. Similarly, the input 7 yields 45, and 2 yields 20. The figure shows the beginning of a computation with Put In −1. The +2 machine is ready to fire. Perhaps some students will predict that the calculation will result in Get Out 5.

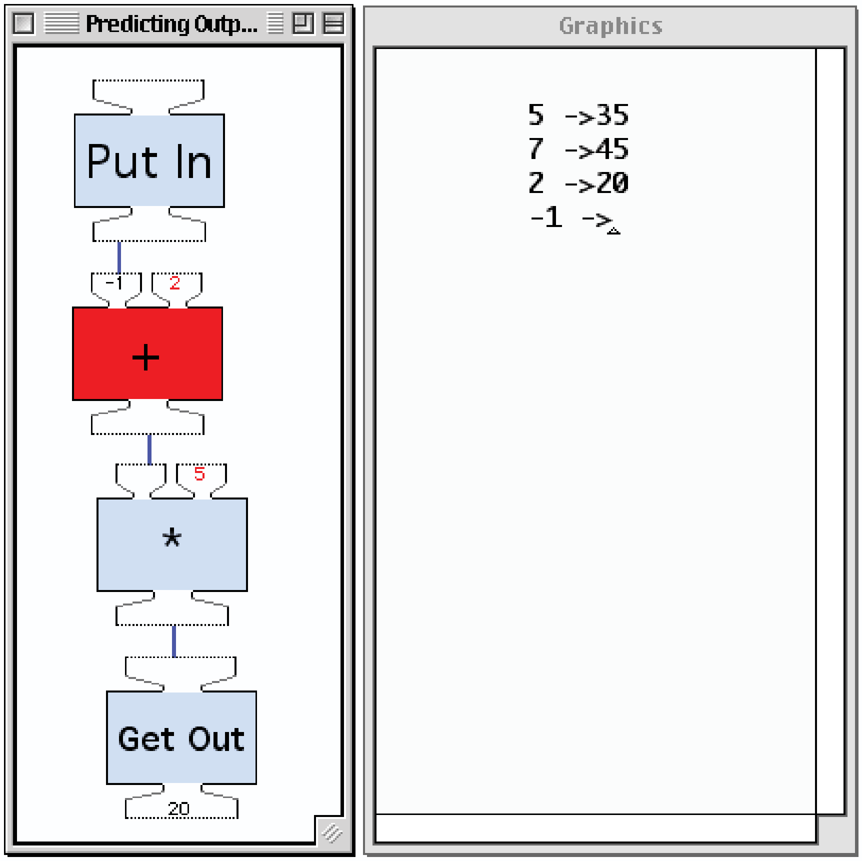


Figure 13.4 Predicting Results *Function Machine.*

The next activity, Guess My Rule, is shown in Figure 13.5. The Put In and Get Out machines are as above. The Mystery machine, however, is new. It has concealed inside it two calculation machines (+, -, *, or / machines).

The activity proceeds as follows. Students are organized into groups of five. Each group has a computer with the Guess My Rule program installed in it. Group A creates two calculation machines and conceals them inside the Mystery machine in Group B's computer. The task of Group B is to run Group A's *Guess My Rule* program with a variety of inputs, and to determine from the resulting outputs, which two calculation machines are inside the *Mystery* machine. At the same time, Group B makes a pair of calculation machines and conceals them inside the *Mystery* machine of Group A, and the kids in Group A try to determine what's inside. This "guessing" game exercises students' thinking about arithmetic operations. Most kids found this challenge a great deal of fun. The figure on the right above shows a session of *Guess My Rule* in progress. The display shows that inputs 0, 1, 2, and 5 produce outputs of 18, 20, 22, and 28 respectively. The current input, 10, is about to be run by the *Mystery* machine. Can the students guess what output it will generate?

Figure 13.6 shows the inner contents of the *Mystery* machine used in the above session- the two machines $+9$ and $*2$. It is not a trivial task for fifth-graders to infer this or other possible solutions. (For example, instead of the machines $+9$ and then $*2$, an equivalent *Mystery* machine is $*2$ and then $+18$.)

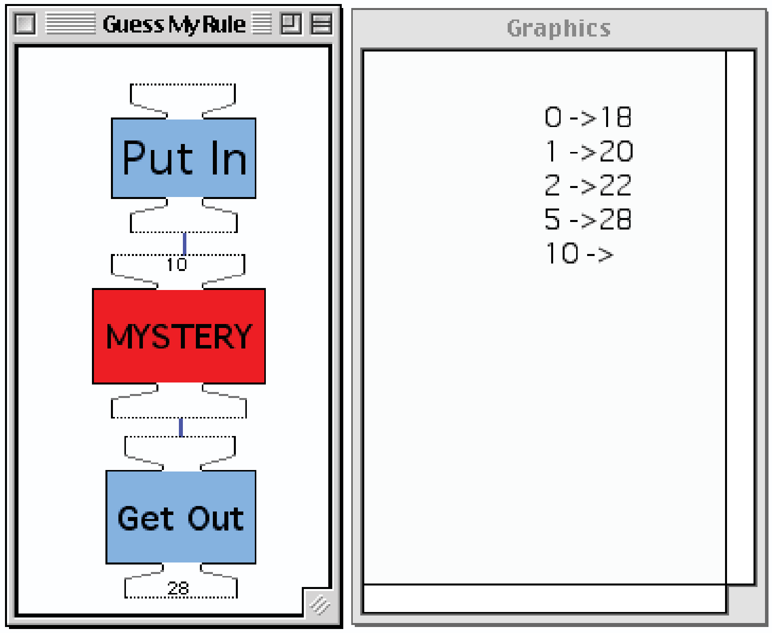


Figure 13.5 Guess My Rule *Function Machine*.

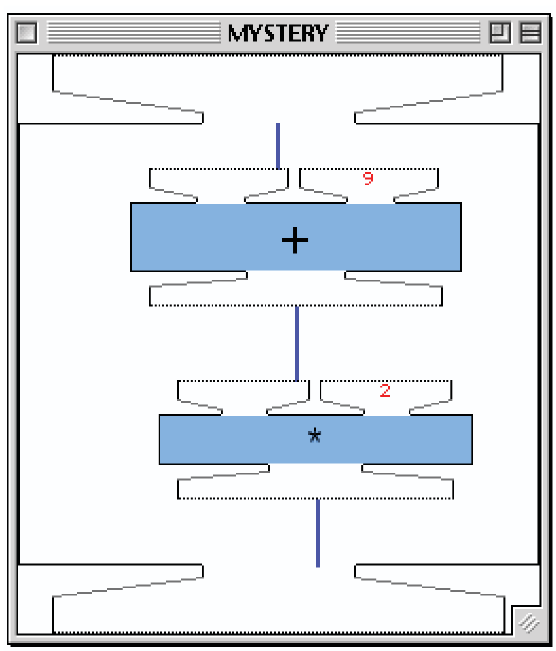


Figure 13.6 Inner contents of the *Mystery* Machine.

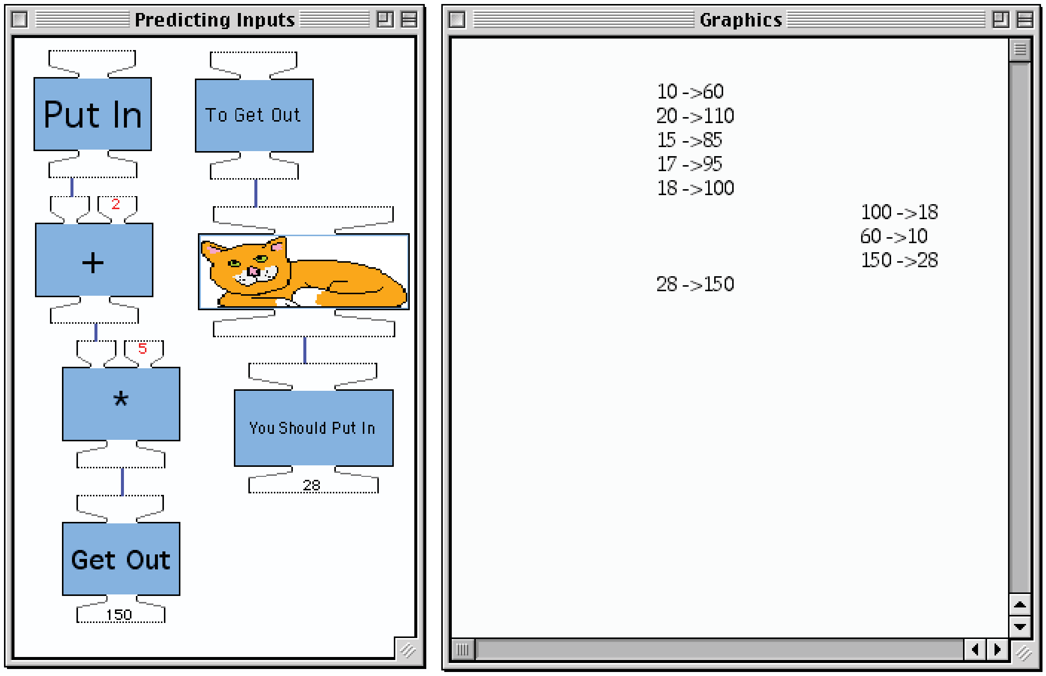


Figure 13.7 Predicting Inputs machine.

Figure 13.7 shows the introduction of a new problem. The left window shows the same sequence of machines used in the first figure in the instructional sequence. However, the new challenge from running these machines, is not to determine what output will be produced by a given input, as previously. *Instead*, the task is to answer the *opposite* question: what number must be input to produce a specified output?

For example, using the given +2 and ∗5 machines, what number must be Put In to Get Out 100? Understandably, this seems to the kids a much more difficult problem. They are encouraged to approach the task by trial and error. The display screen on the right shows the results of a trial-and-error sequence aimed at finding what input yields an output of 100. The input 10 yields 60, so perhaps a larger input is needed. An input of 20 yields 110. So perhaps some input between 10 and 20 is called for. 15 yields 85; 17 yields 95, and then-“Hooray! 18 works!” The sequence required five trials. Using the same pair of machines with other output targets, kids attempt to determine the correct inputs in as few trials as possible. They are delighted when they can zoom in on the answer in three or four trials and sometimes even two!

At this point, the two new machines in the left window, To Get Out and You Should Put In, are introduced. The inputs and outputs printed by these machines are shown on the right half of the display screen. Just after the printout on the left was produced, showing that an input of 18 produces an output of 100, the two new machines are run. An input of 100, the target output in the previous problem, is given to the *To Get Out* machine. This input is piped to the *You Should Put In* machine which produces the printout of 18 shown on the display. Somehow, using this pair of machines, all one had to do was give it the desired output and it produced the corresponding input-and in just a single trial! Also, as the subsequent printout from this machine shows, it confirms what was shown before, that to get an output of 60 requires an input of 10. And it could be used to confirm the other pairs also-that an output of 110 calls for an input of 20, etc. Instead, however, it is used to assert a new claim, that to get an output of 150 requires an input of 28. The last printout on the left of the display, generated by running the four machines on the left, confirms this-an input of 28 indeed yields an output of 150.

The problem for the students is: how does the *You Should Put In* machine know, right from the start, what input one needs to put in to get a desired output? What kind of magic does it use? The kids are not expected to fathom the answer. Instead, they are shown the inner contents of the machine, seen in Figure 13.8.

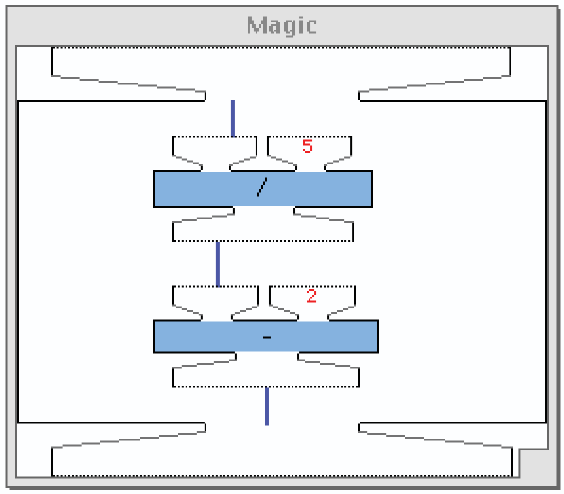


Figure 13.8 Magic machine.

As the figure shows, the solution to the problem is to do the inverse computations in the opposite order. (This is the same algorithm used for turtle path reversal in *Logo*. The original computation sequence was: put in a number (say N), add 2 to it, multiply the result by 5, and get out the answer. To undo this sequence, one computes operations in the reverse order from the original sequence, replacing the original operations with their inverses, as follows. Put in the desired answer, divide it by 5, subtract 2 from the result and get out N. This is essentially the algorithm for solving linear equations in algebra, and in this context, fifth-graders can understand its sense and purpose.

MultiMap

MultiMap is a software tool for introducing experimental mathematics into the pre-college curriculum through investigating the dynamics of planar maps. It aimed to introduce students to the concept of a map, seen as a transformation of the plane onto itself. The *MultiMap* program transforms figures on the computer screen according to rules (i.e., *maps*) specified by the user. Linear maps can be created from primitive operations such as rotation, scaling, and translation. The rules can also be expressed as mathematical functions. Though *MultiMap* is accessible to middle-school students, it also provides professional mathematicians an environment in which they can encounter challenging questions. It was originally designed as a general-purpose tool to support a high-school curriculum in mathematical chaos. It was developed by Horwitz, Feurzeig, and MIT consultant Michael Eisenberg, and implemented on the Macintosh by BBN software developer Bin Zhang.

MultiMap has a direct manipulation iconic interface with extensive facilities for creating maps and studying their properties under iteration. The user creates figures (such as points, lines, rectangles, circles, and polygons), and the program graphically displays the image of these figures transformed by the map, possibly under iteration. *MultiMap* allows one to make more complex maps out of previously created maps in three distinct ways: by composition, superposition, or random selection of submaps. It includes a facility for coloring maps by iteration number, a crosshair tool for tracing a figure in the domain to see the corresponding points in the range, a zoom tool for magnifying or contracting the scale of the windows, and a number of other investigative facilities. *MultiMap* also enables the generation and investigation of nonlinear maps that may have chaotic dynamics. The program supports the creation of self-similar fractals, allowing one to produce figures that are often ornate and beautiful. Using *MultiMap*, and with minimal guidance from an instructor, students have discovered such phenomena as limit cycles, quasi-periodicity, eigenvectors, bifurcation, fractals, and strange attractors (Horwitz and Eisenberg, 1991).

When *MultiMap* is called up, the screen is divided into three windows. The domain window enables the user to draw shapes such as points, lines, polygons and rectangular grids, using the iconic tools in the palette on the left. The range window is used by the computer to draw one or more iterates of whatever shapes are drawn in the domain. The map window specifies the transformation of points in the domain that "maps" them into the range. The user controls what the computer draws in the range window by specifying a mapping rule, expressed in the form of a geometric transformation. The map is to be performed on the entire plane, a user-definable portion of which is displayed in the domain window. For example, the default transformation, called the "identity map," simply copies the domain one-for-one into the range.

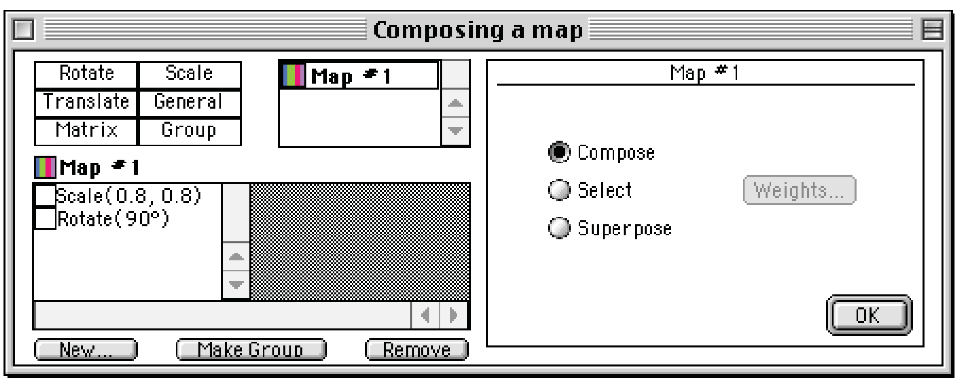


Figure 13.9 Composing a map.

In Figure 13.9, the user has entered a rectangle in the domain window and has then specified a map composed of two submaps, a scale and a rotation. Scale (0.8, 0.8) scales the rectangle to 0.8 of its original size in both x and y. Rotate (90°) rotates the rectangle 90 degrees about the origin. In a composition map such as this, the transformations are performed in order. Thus the rectangle is scaled and then rotated. This is an iterated map. The user has specified that the map is to be performed 4 times (after including the identity map), with a distinct color for successive iterations (light blue, green, red, and pink). The range window shows the result of the mapping.

Using *MultiMap*, students from local high schools created and investigated simple maps built on the familiar operations of rotation, scaling, and translation. They then investigated the behavior under iteration of more-complex maps, including maps that produce beautiful fractals with self-similar features at all levels, random maps that generate regular orderly structures, and maps that, though deterministic, give rise to unpredictable and highly irregular behaviors (chaos). Students were introduced to rotation, scale, and translation maps during their first sessions, and to their properties under composition and iteration.

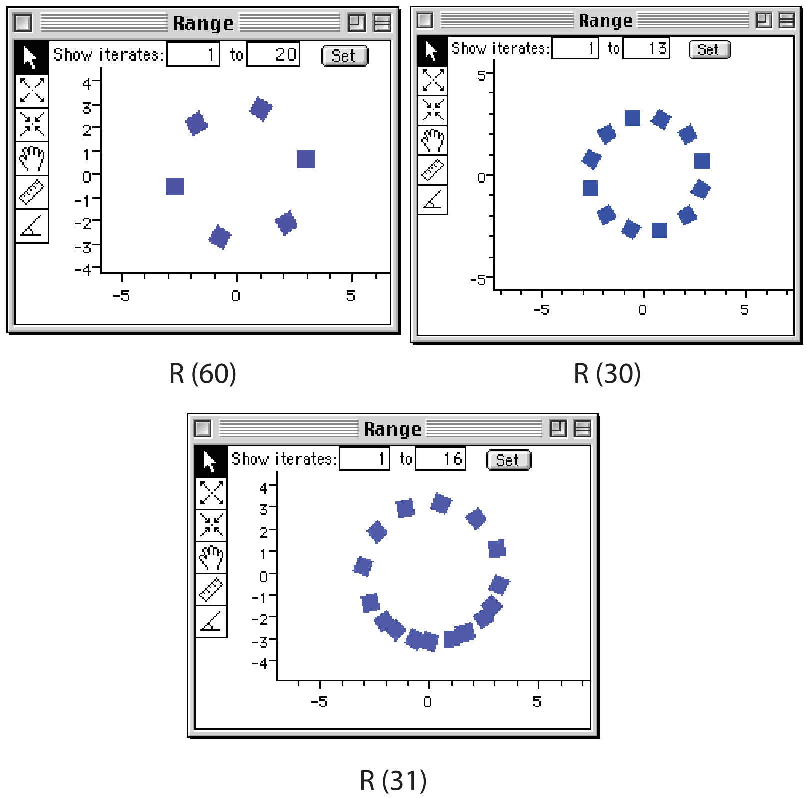
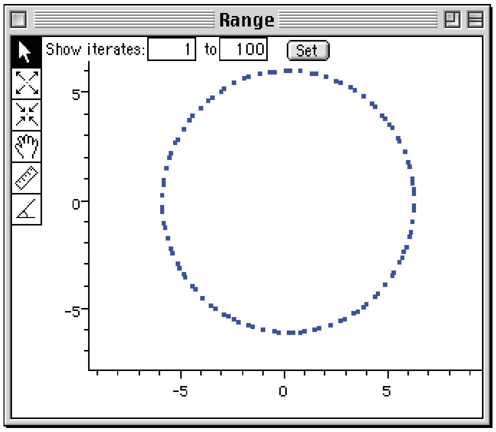


Figure 13.10 A *MultiMap* session.

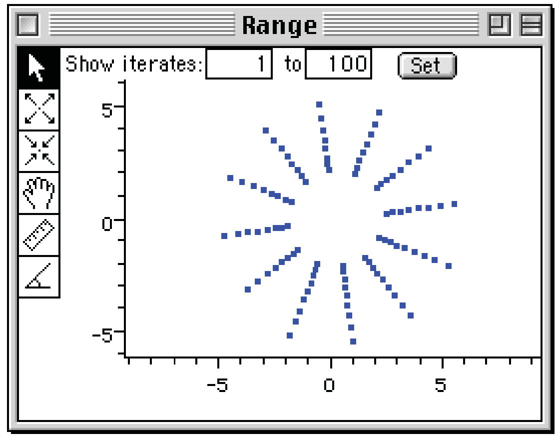
The session shown in Figure 13.10 illustrates the use of *MultiMap* by two students , Kate and Fred, working together on an investigation of rotational symmetry (Horwitz and Feurzeig, 1994). They began by drawing a square and rotating it by 60 degrees, as shown in the top left screenshot. They noted that the 6 copies of the square lay around a circle centered at the origin, and that, though the map was iterated 20 times, after the first 6 iterations the others wrote over the ones already there. They were then asked what the result of a rotation by 30 degrees would be. Kate said that there would be 12 copies of the square instead of 6, no matter how many iterations. They confirmed this, as shown in the top right screenshot. The instructor then asked “What would happen if the rotation angle had been 31 degrees instead of 30?” Fred said “There will be more squares-each one will be one more degree away from the 30 degree place each time, so the squares will cover more of the circle.” *MultiMap* confirmed this, as shown in the bottom screenshot.

Instructor: “The picture would be less crowded if the square was replaced by a point.” Fred made this change. The result, after 100 iterations, is shown on the left of Figure 13.11. Since there was still some overlap, the instructor said “After each rotation let’s scale x and y by .99. That will bring the rotated points in toward the center a little more at each iteration.” Ann then built an R(30°)S(0.99, 0.99) composite map. The effect of the scaling is shown on the right.

Fred said “Now the points come in like the spokes of a wheel with 12 straight arms.”



R (31)



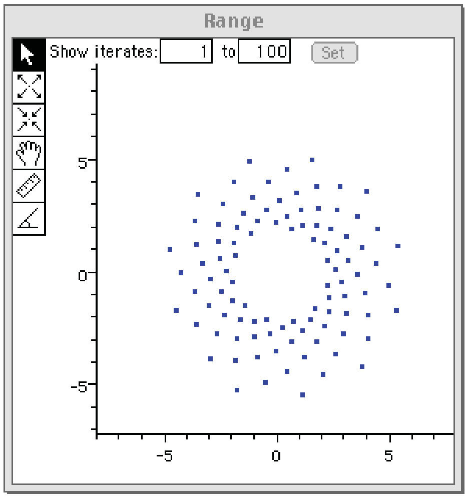
R (30) ∗ S (0.99, 0.99)

Figure 13.11 Continuing the session.

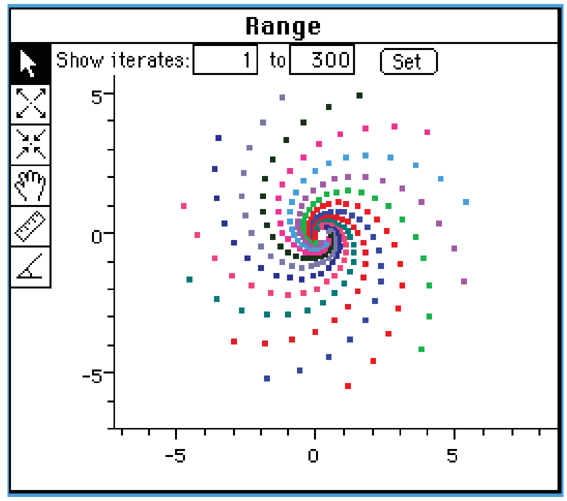
The instructor agreed and asked what would happen if the rotation were 31 degrees instead of 30. Fred replied "It would be almost the same but the points would not be on straight lines. They would curve in a little each time." He tried this. The result is shown in the top left of Figure 13.12. Kate said "The spokes have become spiral arms." When asked how many arms there were, she said "It looks like 12." The instructor said "Let's check that visually by making the points cycle through 12 colors repeatedly so that successive points have distinct colors." The result is shown in the top right of Figure 13.12. Kate: "Oh, how beautiful! And now each arm of the web has the same color." Instructor, "Right, so we can clearly see that the web has 12-fold symmetry."

Instructor: "What do you think will happen if the rotation is 29 degrees instead of 31 degrees?" Kate: "I think it will be another spiral, maybe it will curve the other way, counter-clockwise. But I think it will still have 12-fold symmetry. Here goes!" The result is shown in the middle left of Figure 13.12. Instructor: "Right! It goes counter-clockwise and it does have 12-fold symmetry. Very good! Now let's try a rotation of 27 degrees. What do you think will happen?" Kate: "I think it will be about the same, a 12-fold spiral web, maybe a little more curved." The result is shown in the middle right of Figure 13.12. Instructor: "So what happened?" Kate: "It looks like a 12-fold spiral web but why aren't the colors the same for each arm?". Instructor: "Right! It goes counter-clockwise and it does have 12-fold symmetry."

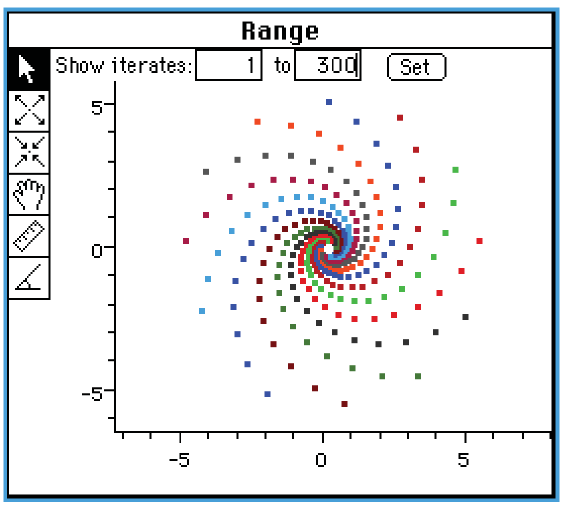
Instructor: "It might be that we don't have enough detail — let's get a more detailed picture by changing the scale from .99 to .999, and increasing the number of iterations from 300 to 600. See if that makes a difference." The result, after 600 iterations, is shown at the bottom left of the figure. Kate: "Wow, it looks very different now! There are many more than 12 arms, but they're all straight, and each arm still has many different colors." Instructor: "There's obviously much more than 12-fold symmetry here. Any idea what it is?" Fred: "120." Instructor: "Why do you say that?" Fred: "Because 360 and 27 have 9 as their greatest common divisor. So 360 divided by 9 is 40, and 27 divided by 9 is 3, and 40 times 3 is 120." Instructor: "What do you think, Kate?" Kate: "I don't know but I counted the arms and it looks like there are 40." Instructor: "Let's see if that's right. Reset the color map so that the colors recycle every 40 iterations instead of every 12 iterations." The students changed the color ramp. The result, after 600 iterations, is shown below on the right.



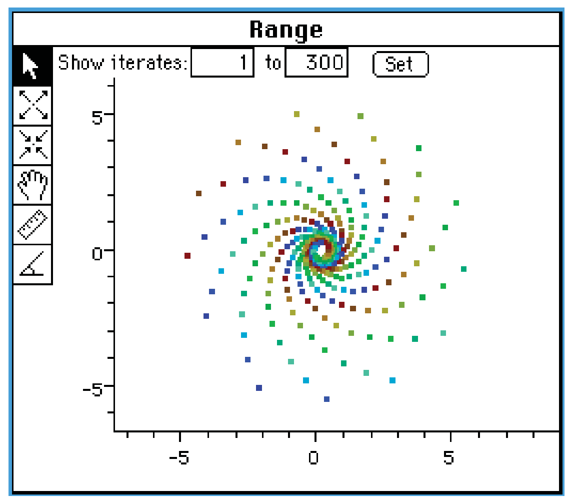
R (31) * S (.99, .99)



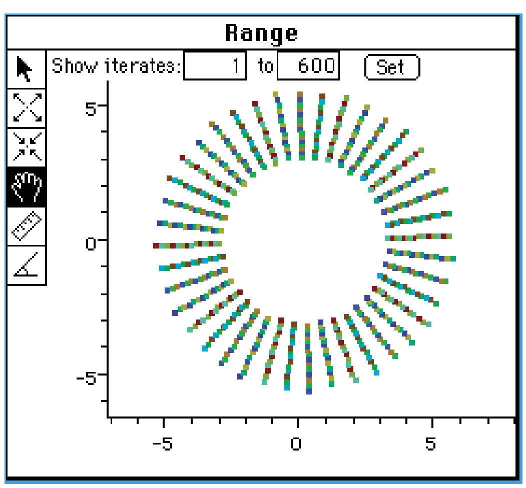
R (31) * S (.99, .99) 12-color ramp



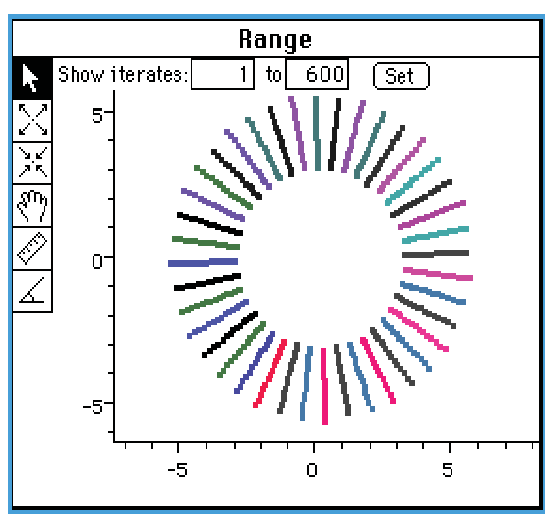
R (29) * S (.99, .99) 12-color ramp



R (27) * S (.99, .99) 12-color ramp



R (27) * S (.999, .999) 12-color ramp



R (27) * S (.999, .999) 40-color ramp

Figure 13.12 More of the session.

Kate: "Now each arm is the same color. So there is 40-fold symmetry." Fred: Is 120 wrong? Instructor: "No, 120 isn't wrong but it's not the only or the best answer. 240 and 360 would work and so would any other multiple of 120. But the real question is: what is the smallest one? The way to view the problem is this: what is the least number of times you have to go around a circle in 27-degree increments to come back to where you started? Or, to put it another way, what is the smallest integer N such that the 27 times N is an exact multiple of 360? The answer is 40 because 40 times 27 equals 1080, which is 3 times 360. No integer less than 40 will work." Fred: "I understand. Now I can do the problem for any angle."

MultiMap was developed in the NSF project "Advanced Mathematics from an Elementary Viewpoint" (Feurzeig, Horwitz, and Boulanger, 1989). Its use enabled students to gain insights in other visually rich mathematical explorations such as investigations of the self-similar cyclic behavior of the limiting orbits of rotations with non-uniform scaling (Horwitz and Feurzeig, 1994).

ELASTIC

The *ELASTIC* software system is a set of tools for exploring the objects and processes involved in statistical reasoning. It was developed for use in a new approach to teaching statistical concepts and applications in a high-school course called Reasoning Under Uncertainty (Rubin et al, 1988). *ELASTIC* supports straightforward capabilities for entering, manipulating, and displaying data. It couples the power of a database management system with novel capabilities for graphing histograms, boxplots, scatterplots, and barplots. *ELASTIC* provides three special interactive visual tools that serve students as a laboratory for exploring the meaning underlying abstract statistical concepts and processes. One tool, Stretchy Histograms, shown in Figure 13.13, enables students to manipulate the shape of a distribution represented in a histogram.

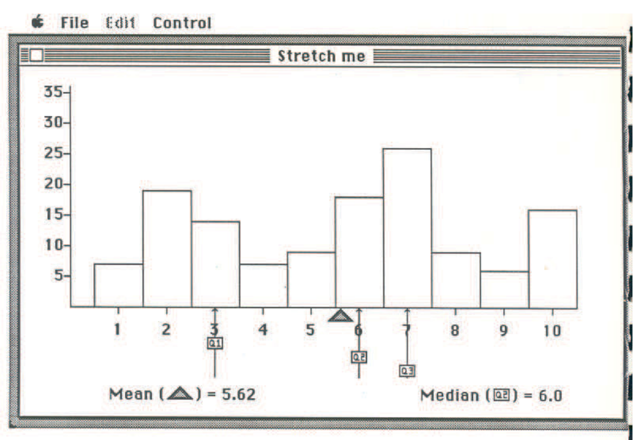


Figure 13.13 Stretchy Histograms.

Using the mouse, they can stretch or shrink the relative frequencies of classes in a distribution and watch as graphical representations of mean, median, and quartiles are dynamically updated to reflect those changes. In this way, students can explore the relationships among a distribution's shape, its measures of central tendency and its measures of variability. They can also use the program to construct histograms that represent their hypotheses about distributions in the real world. Another tool, Sampler, shown in Figure 13.14, is a laboratory for exploring the process of sampling.

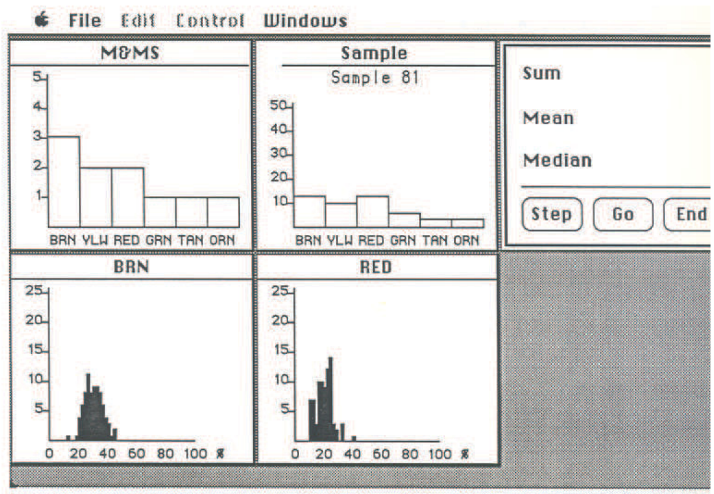


Figure 13.14 Sampler.

A student or teacher can create a hypothetical population and then draw any number of samples of any size from it. Sampler displays graphs of the population model, the sample that is currently being drawn, and summary data on the set of samples, including a distribution of sample means. Students can use Sampler to run experiments, for example by taking repeated samples and watching as the distribution of sample means or medians grows. They can also compare the distribution of samples to real world samples they have generated. A third tool, Shifty Lines, shown in Figure 13.15, is an interactive scatterplot that enables students to experiment with line fitting, adjusting the slope and y-intercept of a regression line, and observing the resulting fit of the line to the data points.

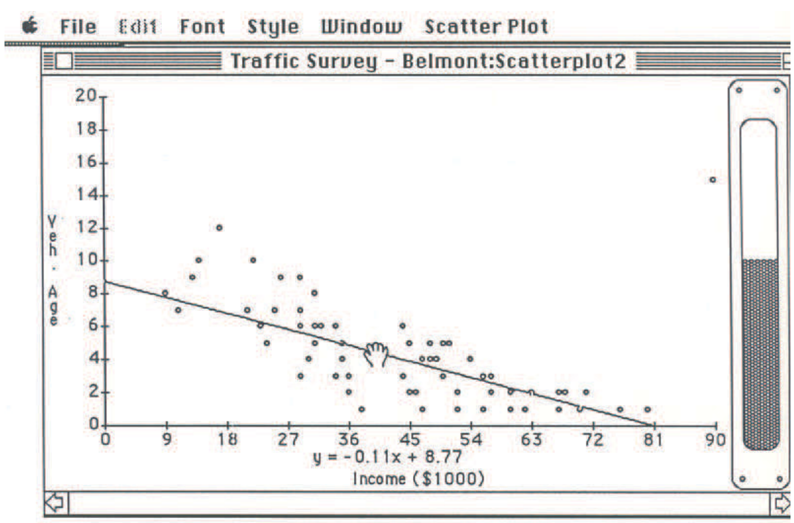


Figure 13.15 Shifty Lines.

The software provides students with several kinds of information: 1) dots for each data point, 2) a straight line that represents an hypothesis about the x-y relationship, 3) a thermometer icon that displays the "goodness of fit" of the line to the points, 4) marks on the thermometer that show the best fit achieved so far and the best theoretical fit, and 5) an equation for the current straight line.

As students adjust the regression line, the sum of squares "thermometer" changes to reflect their actions. They can temporarily "eliminate" points from the scatterplot and watch as the other representations are automatically updated. They can query a point on the graph and receive information about it from the database. Thus, using Shifty Lines, students can explore multiple sources of information about multivariate data.

The project involved the following BBN scientists and support staff. Andee Rubin, Ann Rosebery, Bertram Bruce, John Swets, Wallace Feurzeig, Paul Biondo, Willian DuMouchel, Carl Feehrer, Paul Horwitz, Meredith Lesly, Tricia Neth, Ray Nickerson, John Richards, William Salter, Sue Stelmack, and Sterling Wight. The software was published as the Statistics Workshop by Sunburst Communications Inc. in 1991.

Topology software

From 1995 through 1997, Feurzeig and BBN mathematicians Gabriel Katz and Philip Lewis, in collaboration with consultant Jeffrey Weeks, a topologist and computer scientist, conducted curriculum and software research in the project "Teaching Mathematical Thinking Through Computer-Aided Space Explorations." The object was to introduce some of the most fundamental and central ideas of geometry and topology to a broad population of high-school students through a series of interactive graphical explorations, experiments, and games involving the study of two- and three-dimensional spatial objects. The initial activities included exploratory investigations of the mathematics of surfaces. To help students experience the topology of the torus and Klein bottle surfaces, Weeks developed six software games to be played on these unfamiliar surfaces (Weeks, 1996). Though the games are familiar, playing them when the moves have quite different results from those made in the usual flat world is very challenging. Figure 13.16 shows the screen display for the entry points to the six games.

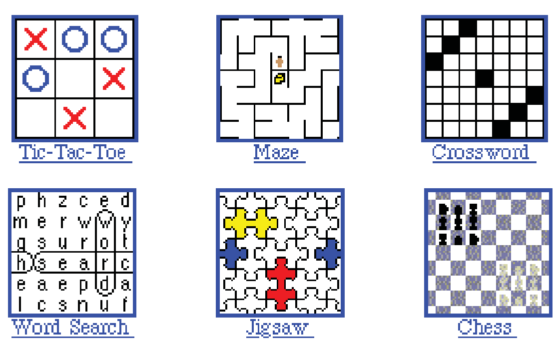


Figure 13.16 Entry points to games.

The illustrations in the figure show the games in the "fundamental domain," as they appear when played on a flat surface. The software enables the games to be viewed as they appear when played on a toroidal or Klein bottle surface. For example, the player can scroll the Tic-Tac-Toe board in the above torus game to see that the X's have won — the third X on the extended toroidal surface is just to the right of the top row, forming 3 Xs in a line. These games are accessible for download at `http://www.geometrygames.org`

Rising Curtain, a software applet, was developed for teaching the Euler formula, which relates the number of vertices (V), edges (E), and faces (F) in planar graphs. In this environment, a web of intersecting straight lines in the plane (the computer screen) is partially covered by a curtain under user control. As a student raises the curtain, local changes (newly appearing vertices, edges, or faces) are revealed. The student's task is to count the number of each of these features as the curtain rises to show the entire graph, and then to determine how these are related, i.e., to discover the Euler invariant for planar graphs ($V - E + F = 1$). Students compute the Euler numbers of a few model surfaces: the torus, Mobius band, and Klein bottle. Rising Curtain is an effective tool for introducing the fundamental mathematical concept of an invariant.

Students were then guided through a gentle, semi-rigorous development of the powerful classification theorem of surfaces, which establishes that all possible surfaces are combinations of a few basic ones, such as the torus and Klein bottle. The final project activities involved computer-simulated journeys in the world of two-dimensional spaces, using *2D-Explorer*, an interactive software tool. This software enables students to explore the surface of an unknown mathematical planet — a closed 2D-surface — with the goal of determining its global structure from local observations. Players piloting a low-altitude "flying machine" undertake voyages to uncharted closed-surface worlds. Given a mystery "planet," they are challenged to answer the question "What is the shape of this universe?" (Weeks, 1997).

Their task, as they travel over the unknown planet is to determine the intrinsic global topology of its surface by making local measurements and observations along the way. The program employs graphically rich textures and 3D animation. However, although it presents a 3-dimensional world, the underlying topological connections are only 2-dimensional. An understanding of the characteristic mathematical structure of different surfaces enables the user to establish the topology of the territory she has explored. This permits her to compute the Euler number of the known part of S. By application of the classification theorem, she knows the topology of the part of S that she has explored thus far, and the possible topologies of S that are not yet ruled out. Then, if one day, pushing the final frontier, she fails to discover new territories, she has visited everywhere and her mission is over: she knows the shape of that universe!

From 1997 through 1999, Feurzeig and Katz, in collaboration with mathematics faculty from four universities, developed versions of a new undergraduate course under the NSF-supported project "Looking into Mathematics: A Visual Invitation to Mathematical Thinking." The universities (Brandeis, Clark, Harvard, and the University of Massachusetts, Boston) were pilot sites for the course. The course included units on visual representations of mathematical objects and universes, mathematical maps, curves and surfaces, and topological explorations of "the shape of space." Visual software treating all these topics was developed to support the teaching. The student populations included pre-service elementary school teachers, in-service high-school mathematics teachers, and non-mathematics majors.. Though the four pilot versions of the course differed somewhat in emphasis, as appropriate for their different populations, they had substantial commonality in content and pedagogic approach.

13.3 Learning and teaching science

Much of our understanding of the workings of the physical world stems from our ability to construct models. BBN work has pioneered the development of computational models that enable new approaches to science inquiry. Several innovative software environments that employ interactive visual modeling facilities for supporting student work in biology and physics are described next.

Thinkertools

This 1984–1987 NSF research project, conducted by physicist Paul Horwitz and cognitive scientist Barbara White, explored an innovative approach to using microcomputers for teaching Newtonian physics. The learning activities were centered on problem solving and experimentation within a series of computer microworlds (domain-specific interactive simulations) called *ThinkerTools* (White and Horwitz, 1987; Horwitz and White, 1988). Starting from an analysis of students' misconceptions and preconceptions, the activities were designed to confront students' misconceptions and build on their useful prior knowledge.

ThinkerTools activities were set in the context of microworlds that embodied Newton's laws of motion and provided students with dynamic representations of the relevant physical phenomena. The objective was to help students acquire an increasingly sophisticated causal model for reasoning about how forces affect the motion of objects. To facilitate the evolution of such a model, the microworlds incorporated a variety of linked alternative representations for force and motion, and a set of game-like activities designed to focus the students' inductive learning processes.

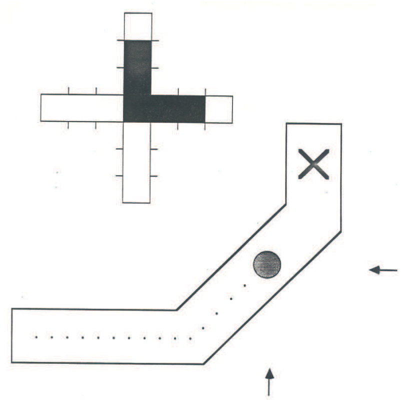


Figure 13.17 *ThinkerTools* game.

Figure 13.17 displays a typical *ThinkerTools* game. The user tries to control the motion of an object so that it navigates a track and stops on the target X. The shaded circle in the middle of the angled path is the object, which is referred to as the "dot." Fixed size impulses, in the left-right or up-down directions can be applied to the dot via a joystick. The dot leaves "wakes" in the form of little dots laid down at regular time intervals. The large cross at the top is the "datacross." This is a pair of crossed "thermometers" that register the horizontal and vertical velocity components of the dot, as indicated by the amount of "mercury" in them. Here the datacross is depicting a velocity inclined at +45 degrees relative to the horizontal. Sixth-grade students learned to use the datacross to determine the dot's speed and direction of motion.

As part of the pedagogical approach, students formalized what they learned into a set of laws which they examined critically, using criteria such as correctness, generality, and parsimony. They then went on to apply these laws to a variety of real world problems. Their investigations of the physics subject matter served to facilitate students' learning about the acquisition of scientific knowledge in general — the nature, evolution, and application of scientific laws. Sixth-grade students using a sequence of *ThinkerTools* problems did better on classical force and motion problems than high-school students using traditional methods.

Explorer Science models

The *Explorer Science* series combined the use of analytical capabilities with scientific models to create simulations for learning physics and biology. The software was developed by BBN scientists John Richards and Bill Barowy with Logal Educational Software, Israel (Barowy, Richards, and Levin, 1992). Animated measuring and manipulation tools complemented the dynamic simulations. Analytic capabilities included graphs, charts, and an internal spreadsheet with automatic or manual data collection. The *Physics Explorer* and *Biology Explorer* software was published by Wings for learning.

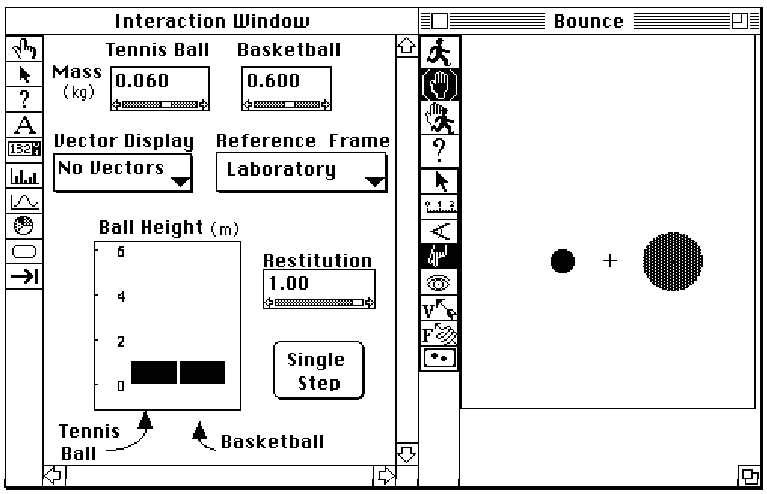


Figure 13.18 Tennis ball and basketball interaction.

The example in Figure 13.18, taken from physics classroom work, illustrates how computer modeling was integrated with laboratory experimentation to foster a coherent approach to science inquiry. The use of the model facilitates analysis and conceptual understanding of the physical phenomena. By dealing explicitly with differences between computer models and the phenomena they simulate it was possible to engage students in fruitful discussions about the strengths and limitations of the models. The students developed a sense of how scientists use models by trying to simulate phenomena themselves.

The example is an exploration of how a tennis ball and a basketball interact when the two are dropped at the same time with the tennis ball directly above and close to the basketball. The class observed both the real phenomena and the simulation with the *Explorer* Two-Body model. In successive stages of the inquiry, the focus of attention alternated between the actual phenomena and the simulation. In the first experiment, the basketball and the tennis ball were held side by side, at about chest height. The instructor asked the students to predict what the height of the first bounce of each ball would be when the two are dropped together. They were asked to explain the reasons for their predictions in order to make their intuitions explicit. After students responded, the experiment was performed. The tennis ball and the basketball each bounced to a height about 3/4 of that from which they had been dropped, which occasioned little surprise. This initial experiment established a baseline observation for checking the credibility of the computer model when it was used to replicate the experiment. The figure shows the lab that was designed to simulate the experiment. The Two-Body model is a mathematical representation of time-dependent interactions between two circular objects and four stationary walls. The animation generated from the model appears in the model window to the right. A work-window, the Interaction Window, is shown to the left.

Using the model, students investigated the effect of changing the coefficient of restitution. The two real balls were weighed and the masses of the objects in the simulation were adjusted to match. This stage of the investigation focused on the relation of the model to the phenomena. The class discussed how they would determine when they had found a satisfactory simulation.

In the next stage of the activity, the real basketball was held at chest height with the tennis ball about 5 centimeters directly above it. The students were challenged again to predict what would happen when the balls were dropped. Would the basketball bounce as high as it did before? What about the tennis ball? They discussed their ideas and committed their predictions to paper. After several minutes of discussion, when the members of the individual teams had reached a general consensus, the balls were dropped together. Most students predicted that the tennis ball would bounce no higher than the instructor's head. When it is dropped above the basketball from shoulder height, the tennis ball often bounces up to 15 feet in height, and dramatically strikes the ceiling. Students were surprised to find that it bounced much higher then they had expected. They wanted to know why.

The analytic solution to the problem involves solving simultaneous equations, which was beyond the abilities of the students in the class. The computer model, on the other hand, provides the analytic tools to help students acquire a semi-quantitative understanding of the processes that give rise to the phenomena. The experiment was recreated with the software. Two semi-quantitative explanations were developed to account for the phenomena. Both were investigated using the software tools. Both gave reasonable estimates for the maximum height of the tennis ball bounce, though they used different physical principles and *Explorer* tools.

The first explanation was based on energy conservation: whatever energy the tennis ball gains in the collision must be lost by the basketball. This results in a relation between the change in the bounce height of the basketball (ΔH) and the tennis ball (Δh). The relation is $\Delta h/\Delta H = M/m$, the ratio of the mass of the basketball to that of the tennis ball. The change in height for each ball is determined by measuring the distance from the height it was dropped to the maximum height it attains after the interaction. The second explanation uses the principle of momentum conservation and transformations between different frames of reference. This solution was used to supplement the first, to illustrate the possibility of multiple solutions to a problem. By pressing the Single Step button several times, the instructor broke down the motion to enable a frame-by-frame analysis. The sequence of frames is shown in the composite strobe-like diagram of Figure 13.19.

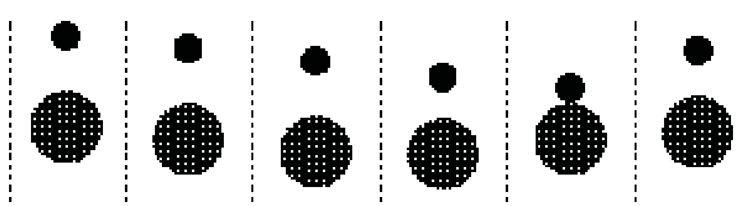


Figure 13.19 Sequence of frames.

Frame-by-frame observations showed that the basketball bounces off the floor first and then, while moving upward, it collides with the tennis ball. Pop-up menus in

Explorer allowed the user to show vector representations for the objects' velocity and acceleration and their components. The students observed that, at the moment of collision, the basketball is moving upward with the same speed as the downward moving tennis ball. Since the basketball is more massive than the tennis ball, it is virtually unaffected by the collision. Therefore, in the laboratory frame, it continued to move upward with approximately the same speed.

Next, the collision between the two balls was viewed from the reference frame that moves upward with the pre-collision speed of the basketball. Just before the collision, the basketball is stationary and the tennis ball approaches with a velocity equal to the difference of the individual velocities. Thus, in this frame, the tennis ball is moving with twice the speed it has in the laboratory frame. The tennis ball rebounds from the basketball with the same speed that it approached the basketball. By transforming back to the laboratory frame, the students estimated that the speed of the tennis ball just after the collision was equal to its speed with respect to the basketball plus the speed of the basketball with respect to the lab. Its speed was greater than it was just before the collision by a factor of three. Students gained an understanding of the surprising result — the "kick" the basketball gave to the tennis ball — and more importantly, familiarity with the use of a powerful science inquiry tool.

The *Explorer* science series has been used in hundreds of school systems throughout the United States. *Physics Explorer* has won numerous awards, including a prestigious Methods and Media Award in 1991.

RelLab

RelLab was developed by Paul Horwitz, Kerry Shetline, and consultant Edwin Taylor under the NSF project Modern Physics from an Elementary Viewpoint (Horwitz, Taylor, and Hickman, 1994; Horwitz, Taylor, and Barowy, 1996). It presents students a computer-based "relativity laboratory" with which they can perform a wide variety of gedanken experiments in the form of animated scenarios involving objects that move about the screen. *RelLab* enables students to create representations of physical objects in the form of computer icons and then assign them any speed up to (but not including) the speed of light. If they wish, they can instruct their objects to change velocity or emit another object at particular instants during the running of the scenario. At any time, as they are building their scenario, the students can run it and observe its behavior in the reference frame of any object. A representative *RelLab* screen is shown in Figure 13.20. The objects in the scenario are a football player and a rocket.

The football player is running at four meters per second. If the animation were run, the icon would move from left to right across the screen, taking approximately 12 seconds to traverse it. The rocket is moving up the screen at two-thirds the speed of light. If the animation were run at normal speed, it would disappear instantly off the top of the screen. Both the football player and the rocket have been given clocks that measure the time in their respective reference frames. The football player's clock matches that of the current frame, but the rocket's shows a different time. This relativistic effect is a fundamental consequence of the constancy of the speed of light, and the reason for it is one of the hardest things for students to learn.

Since relativity deals with time and space, a major consideration in designing *RelLab* was to build into it comprehensive but easily understood representations of these quantities, as well as powerful ways of manipulating and measuring them. All the concepts we wanted to teach could be handled as easily in two dimensions as in three. Thus, every scenario in *RelLab* is viewed as it would appear either from a helicopter looking down on the scene, or horizontally out the window of a train or car moving at

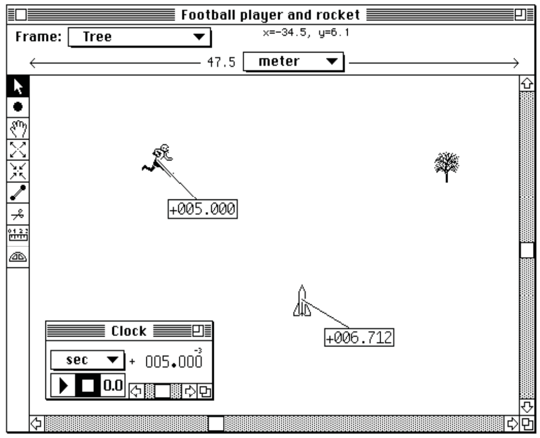


Figure 13.20 Representative *RelLab* screen.

the same speed as the reference object. *RelLab* scenarios often involve astronomical distances. Since there are no obvious indications of the space scale (the icons, which represent point objects, do not change size when the screen is zoomed), this can be confusing to students. Thus, *RelLab* provides a continuously available indication of the space scale, in the form of arrows that span the top of the screen and indicate (in units selectable by the student) how far it is across.

Time is represented directly in *RelLab* through animation. As a result of another explicit design decision, *RelLab* does not allow the user to alter the rate at which time passes: there are no time-lapse or slow-motion displays. When animated, every scenario runs in "real time" — one second on the computer being exactly equivalent to one second in the scenario itself. Early in the design of *RelLab* it was decided not to allow students to build simulations in which the speed of light is altered. This was done not only to avoid possible confusion, but also because such a fundamental alteration of the laws of physics would lead to internal inconsistencies. Instead, *RelLab* demonstrates relativistic effects, which are ordinarily too small to be observed, either by allowing objects to move at very high speeds or by enabling students to make extremely precise measurements of low-speed scenarios. This, in turn, required us to provide exceedingly fine measuring tools, and indeed *RelLab* allows students to measure distances, such as the separation between objects, and time intervals to a precision limited only by that of the computer.

In addition to representing and measuring time and space, *RelLab* provides students with powerful tools for manipulating these quantities. The *RelLab* screen may be scrolled effectively any amount in any direction, and its width may be set to represent any distance from a few millimeters to many light-years. Time can also be set to any value. The clock that displays time in the current reference frame also gives students control over that time. Any alteration in the time displayed by this clock generates an immediate update of the positions of all objects. By observing such changes on the screen, for example, students can determine the distance traveled during a nanosecond (a billionth of a second) by a rocket traveling at nearly the speed of light.

Representations can be as important for what they conceal as for what they display. For instance, *RelLab* does not represent extended objects; each icon is simply a graphic representation of a single point. The reason for this constraint stems directly from the nature of relativity itself. A rigid object (one that retains its spatial configuration when its velocity changes) is an impossibility in relativity, in part because the speed of sound in such an object would exceed that of light. But students are accustomed to thinking of any extended (solid) object as infinitely rigid, in the sense that an impulsive force applied to one side is transmitted instantly across it. This sort of "pre-relativistic thinking" is likely to lead to confusion, so *RelLab* does not admit the construction of objects that have finite spatial extent. It does, however, allow one to associate point objects that have a semantic association (for instance, the front and back ends of a lance carried by a relativistically galloping knight). Objects of this kind may be connected by drawing straight lines between them. The lines are drawn in gray, however, to convey the fact that they do not correspond to anything physical, and they stretch or shrink if the separation of their endpoints changes. They are analogous to the fictitious lines often drawn between stars to represent the constellations — they denote logical groupings of fundamentally independent objects and imply nothing about the presence of forces between them.

Just as representations can constrain students' emerging conceptions, design choices may entail a conscious decision not to allow students to perform a particular manipulation. One example in *RelLab* is the constraint that objects may not be assigned speeds exceeding the speed of light, conventionally designated by the lowercase letter "c." Attempts to do so bring up an error box. This may seem an arbitrary limitation to students, but their annoyance may lead them to discover that it is not as arbitrary as it may appear. For example, an enterprising student who attempts to bypass it by firing a high-speed projectile from the nose of a rocket moving at close to the speed of light soon discovers that this does not work. No matter how fast the projectile moves (provided it is less than c) in the reference frame of the rocket, its speed never exceeds c in the frame in which the rocket is moving. Relativistic velocities do not add as ordinary ones do.

Another, more subtle constraint arises from the inability of *RelLab* to express cause-and-effect relationships in terms of action at a distance. Every change in a *RelLab* scenario takes place at an event — a particular point in space and time — and has consequences that are local in the sense that they affect only objects in their immediate vicinity. The scripting language that underlies the definition of events in *RelLab* does not allow "if — then" constructions that imply instantaneous action at a distance. For example, a command such as "When the light reaches Andromeda, launch the rocket from Earth" is impossible to express in *RelLab*. Such commands are improperly posed because they imply simultaneity of spatially-separated events and thus can be carried out only in a special subset of reference frames. This frame dependence of simultaneity is a very subtle and completely counter-intuitive concept — perhaps the hardest one for students of relativity to accept and understand. *RelLab* does not explicitly teach this, but because it is built into the very syntax of the event language, the program constrains students to think in purely local terms and prevents them from constructing improperly-posed cause-and-effect relationships.

RelLab won two awards in the 1992 EDUCOM national educational software competition: one for Best Natural Science Software (Physics), the other for Best Design. Using *RelLab* in the classroom, teachers have found that high school students can achieve a qualitative understanding of relativity comparable to that of graduate students.

GenScope

GenScope was developed by Paul Horwitz, Eric Neumann, and Joyce Schwartz from 1993-1996 as the key tool in the NSF project Multi-Level Science (Horwitz, Neumann, and Schwartz, 1996). The goal of the project was to give middle-school and high-school students an understanding of the reasoning processes and mental models that characterize geneticists' knowledge, together with an appreciation for the social and ethical implications of recent advances in the field. *GenScope* is an open-ended computer-based exploratory environment for investigating the phenomena of genetics at several levels (DNA, gene, cell, individual organism, family, and population) in a coherent and unified fashion. Each level offers visual representations of the information available, as well as tools for manipulating that information. The information flows between the levels, linking them in such a way that the effects of manipulations made at any one of them may instantly be observed at each of the others. The levels thus combine to form a seamless program. The software presents the complex, linked, multi-level processes of genetics visually and dynamically to students, making explicit the causal connections and interactions between processes at the various levels. The underlying genetic model is itself linked, via a software structure called a "hypermodel," to a variety of data objects, including video sequences of cell division, visualizations of protein and DNA structure, and organism phenotypes.

To illustrate genetic phenomena, *GenScope* starts with dragons — simple, fictitious creatures that are useful for teaching purposes and do not prematurely raise such sensitive issues as the pros and cons of genetic engineering or the uses of genetic screening tests. Two *GenScope* dragons are shown in Figure 13.21. Students are introduced to

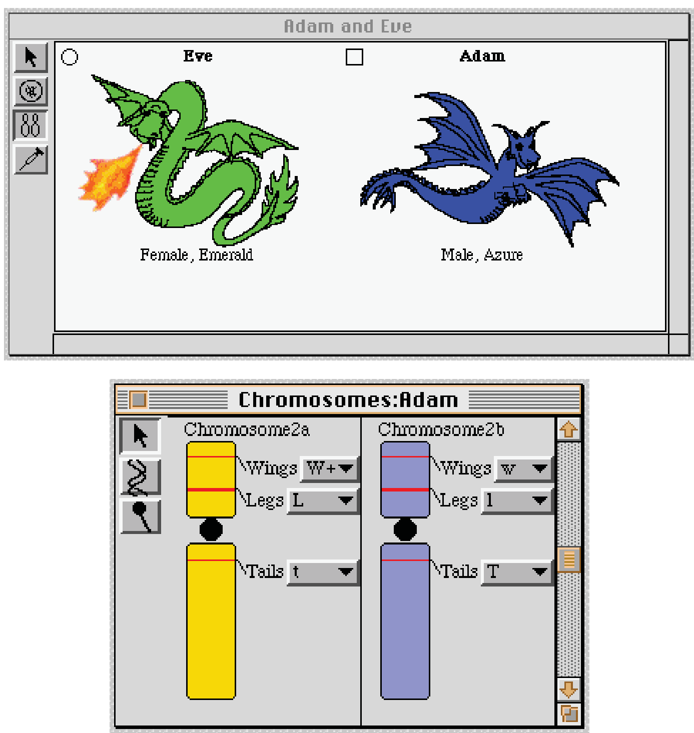


Figure 13.21 *GenScope* screens.

GenScope at the organism level, which displays the organisms' phenotypes (physical traits), but gives no information on their genetic makeup. Using a *GenScope* tool, however, they may move to the chromosome level to observe a pair of chromosomes for an organism, as shown at the bottom of the figure.

The chromosomes, in turn, are made up largely of DNA, which is observable at *GenScope*'s molecular level and carries the genetic information of the particular organism. Genes may be manipulated at either the chromosome or the DNA level, and the results of such manipulations, if any, are immediately observable in the affected organism.

When two organisms mate, their genes are shared by their offspring through two processes which take place at the cell level. The cells can be made to undergo either mitosis, in which process they simply reproduce themselves, or meiosis, whereby they produce a new kind of cell, called a gamete, which possesses only half the chromosomes of the parent cell. The gametes produced through meiosis can then be combined in the central panel of the window, to produce a fertilized cell containing the usual complement of chromosomes. This cell, in turn, will grow into a dragon possessing genetic material from each of its parents. Each of these processes is represented graphically by the computer, as shown in the two illustrations in Figure 13.22. The first figure shows one cell each from Eve and Adam, the two dragons. The spaghetti-looking things in the centers of the cells are chromosomes, the carriers of the genetic information within the cell.

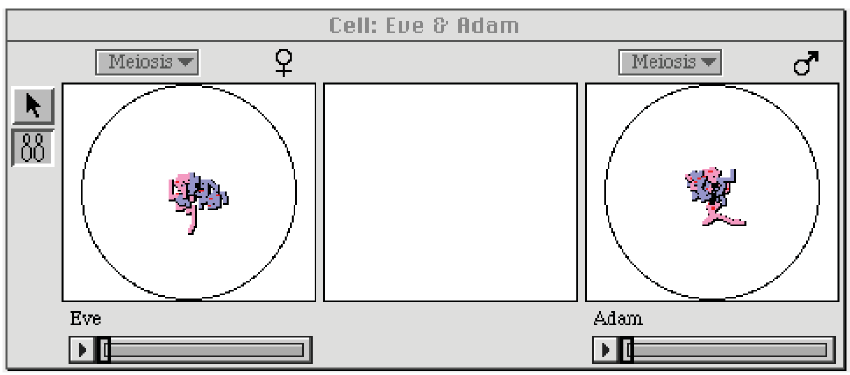


Figure 13.22 Example of mating organisms.

These cells can be made to undergo meiosis (division into four gametes, each of which contains only half the genetic material of the parent cell) or mitosis (ordinary cell division into two identical cells. When meiosis is invoked, the computer runs a randomized simulation of gamete formation. In Figure 13.23 a snapshot of the cell window, meiosis is in process. Adam's cell, on the right, has already produced the four gametes; Eve's cell, on the left, has completed the first division and is halfway through the second.

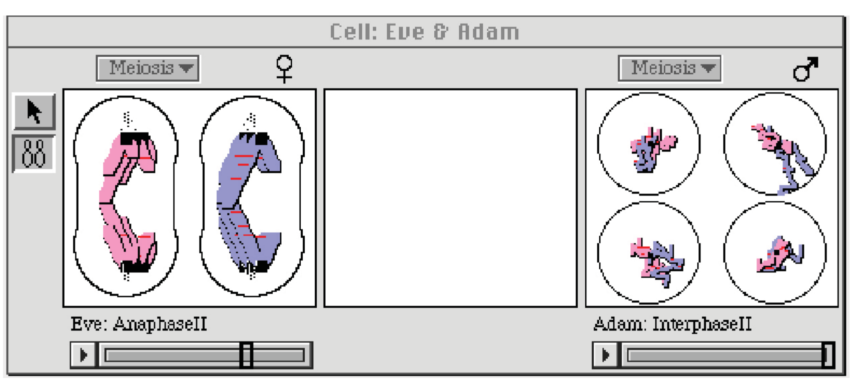


Figure 13.23 Meiosis proceeds.

At *GenScope*'s pedigree level students create "family tree" structures of related organisms in order to observe and investigate such inheritance patterns. The population level displays groups of organisms moving about and randomly mating. Different portions of the screen can be assigned different "environments," which selectively favor one or another phenotype. The resulting "genetic drift" alters the distribution of gene types in the environments. The true nature of the genetic mechanism resides at the molecular level. *GenScope* enables students to drop down to this level to explore the DNA molecule that resides within each chromosome. For example, Figure 13.24 depicts Eve's two genes for wings, showing what the W+ and w alleles look like at the DNA level. The left window shows the dominant and wild type (normally found in the population)

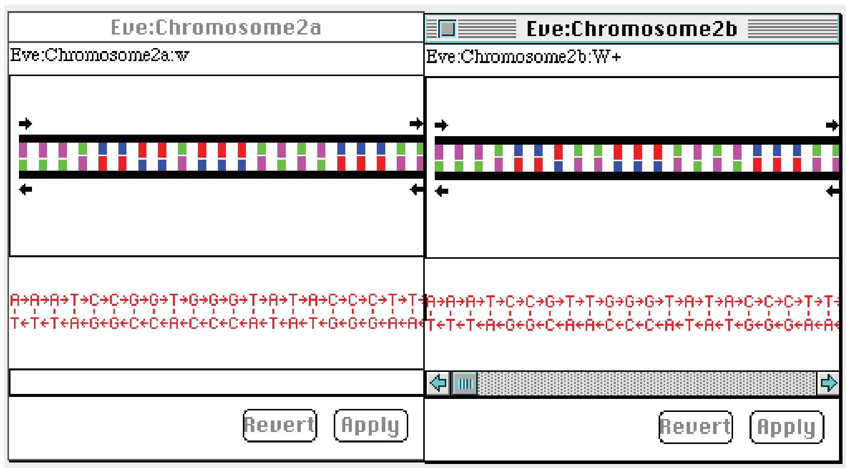


Figure 13.24 More about Eve.

W+ allele, the right window the recessive w allele. They differ by a point mutation — a single base pair substitution.

Just as the informational representation of a gene can be manipulated, via pulldown menus, so the information representation of a DNA strand can also be altered, simply be deleting or inserting the appropriate letters, typing them in as one would do with a word processor. Thus, alleles can be altered at the DNA level, and the changes will be reflected in the organism just as though the gene had been changed directly on the chromosome through the pulldown menu. Mutations created at the DNA level are treated as new alleles. They can be named and used just as the pre-defined ones can.

GenScope was designed to induce students to think at multiple levels. It does this by offering them a set of increasingly difficult challenges, and by careful choice of the set of things they can see and things they can do. On the very first day of class, for example, students are formed into pairs, and issued a simple challenge: "by directly manipulating its genes, try to make this dragon blue, with two legs and no horns." This requires them to explore and experiment with a dominant/recessive trait (horns), a co-dominant one (legs), and a sex-linked, polygenic one (color). Initially, they are given the ability to manipulate the genes directly. In a later activity, the students may be asked to turn their two-legged dragon into a four-legged one, but now their ability to alter the genes at the chromosome level has been disabled. This forces them to work (and therefore to think) at the DNA level, carefully observing the difference between two genes, and then altering one of them appropriately to accomplish the assigned task.

Cardio

Cardio was one of several visual models for science investigation developed in the 1989-1992 NSF-supported project Visual Modeling: A New Experimental Science (Feurzeig, 1994b). BBN project staff included Wallace Feurzeig, Paul Horwitz, John Richards, Ricky Carter, Barry Kort, Eric Neumann, and Donald Redick. *Cardio* is an interactive visual simulation environment for investigating the physiological behavior of the human heart while providing insight into the dynamics of oscillatory processes — particularly coupled oscillators which are fundamental to the operation of living systems. *Cardio*, which was developed by computational biologist Neumann, focuses on the heart's electrical system.

Cardio was designed to permit students to observe the deterministic heart dynamics produced by the cardiac electrical system and to study the effects of changes to specific heart component parameters. As the simulation runs it generates several displays. The major schematic display is the graphic animation of the heart model which shows in real time the rhythmic pulsation of the heart chamber accompanied by the sound of the closing of the heart valves. Another display shows the electrical schematic diagram of the pacemaker nodes and conductance paths. These interactive displays can be simultaneously viewed with EKGs and phase plots showing heart dynamics. During a simulation, the student can record and plot various time-dependent dynamic variables, including EKGs and chamber contraction. This is useful in comparing the dynamics of systems with different parameter values. The *Cardio* screen is shown in Figure 13.25.

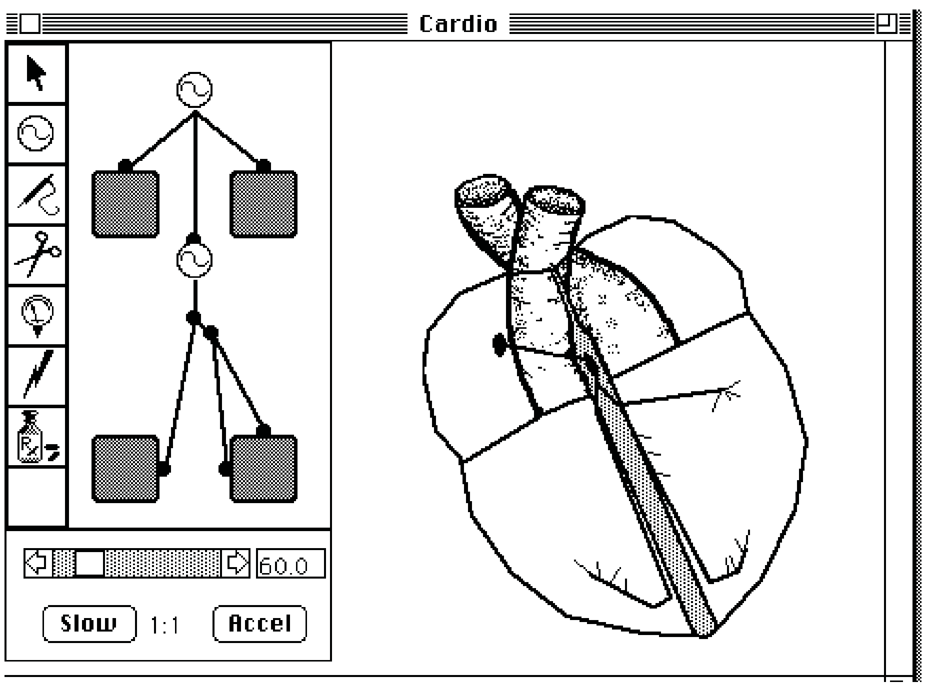


Figure 13.25 *Cardio* screen.

The system dynamics result from the run-time interactions of its individual components. These components include pacemaker nodes, conductance paths, and heart chamber muscle. Students can select from several pre-defined heart dysfunctions and dysrhythmias which alter the component parameters and investigate the resulting dynamics. Because the heart model derives its behavior from component interaction, students can change the parameters of any component or add their own components to form ectopic pacemakers and anomalous conduction paths. In fact, *Cardio*'s visual modeling environment enables students to graphically create new components by selecting them from a palette, specifying their parameters, and connecting them to existing components. This is made possible by using new instances of pre-compiled objects and inserting them into the component list, circumventing the use of a slower and less efficient interpretive structure. Thus, students can create their own heart models and investigate their behaviors.

EKG graphs are constructed and displayed in real time from the 3-dimensional dipole field generated by the four chambers. The depolarization wave of the myocardium creates a positive deflection on the EKG trace as the wave approaches a lead, a negative deflection as the wave recedes, and no deflection if the wave moves orthogonally to the lead. The "L" leads represent the difference between each pair of Einthoven's triangle vertices (i.e., right arm, left arm and feet). Based on the interpretation of at least three different EKG leads, sophisticated users are able to reconstruct the 3-dimensional electric vector time-dependent sweep of the heart. Conduction delays between the atria and ventricles will appear on the tracings as delays between the deflections. EKGs are useful in identifying pacemaker characteristics, conduction rate changes and myocardium anomalies (e.g., ischemia and infarcts).

However, because EKGs are the result of the combined electric fields of each chamber, it is not easy to elucidate from EKG plots the complex and asynchronous patterns of chamber depolarization that continuously evolve over time. Such patterns may arise when the heart does not return to the same state-space after a single pacemaker cycle (as is the case for myocardium which is still in the refractory state caused by the previous pulse). To help visualize such complex behavior, phase plots of the contractions or electric fields of one chamber plotted against those of another chamber illustrate the dynamics by means of orbit paths. For instance, a plot of right atrium contraction vs. left ventricle contraction shows a limit cycle whose eccentricity depends on the phase difference between the two chambers. Complex dysrhythmias can be generated, observed and analyzed in this user-defined phase-space. For example, a second-degree block introduced by the user results in ventricle rhythm that does not always follow the sinus pacemaker, producing multiple orbit paths like those shown in the Figure 13.26 phase plot.

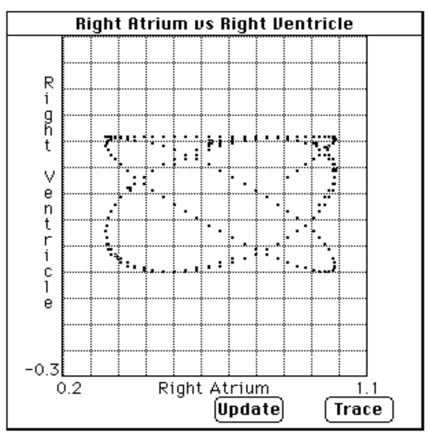


Figure 13.26 Phase plot.

Other types of dynamics can also be created and studied in *Cardio*. Mechanical, electrical, and chemical disturbances of many kinds can be introduced and their effects on heart behavior observed and analyzed. Multiple (i.e., ectopic) pacemakers can be modeled in several ways (e.g., resetting and non-resetting). Combined with the intrinsic refractory limit of the conduction system, these yield complex echo and skip beats.

By saving heart parameters as files, *Cardio* enabled students to compare, model and test various heart conditions and to determine the state-space domains of complex and chaotic rhythms.

Object-Object Transformation Language (OOTLs)

OOTLs is a visual modeling environment for describing dynamic phenomena (Neumann and Feurzeig, 1999). It was developed by Eric Neumann, Wallace Feurzeig, and consultant Peter Garik to help students acquire experience and skill in formulating problems involving dynamical processes. Events in *OOTLs* are conceptualized as interactions among the key objects in the model processes. The *OOTLs* language supports the description and simulation of phenomena for which the law of mass action holds. It applies to "well-stirred" systems composed of large numbers of dynamically interacting objects.

OOTLs has application to an extensive variety of phenomena in many areas of science including epidemiology (contagious disease spread), population ecology (competition, predation, and adaptation), economics (market dynamics), physics (gas kinetics), chemical dynamics (reaction-diffusion equations) and traffic flow. It provides students with a parser to construct equations describing interactions between objects. The objects, which are represented as graphic icons, may represent chemical species, gas molecules, or humans. Objects interact with each other at specified rates. The equations describe the transformations resulting from the object interactions. Objects may be created or consumed (e.g., for chemical reactions there are sources and sinks for reactants; for a biological problem, birth and death of species; for a model of an economy, imports and exports, or innovation and obsolescence). Equations are specified simply, by dragging graphic icons into windows.

OOTLs enables students to study the time behavior of the reactions before they have the mathematics necessary to understand the underlying differential equations. The number of coupled reactions and the number of participating objects are not limited. Objects are assigned arbitrary colors — red, blue and green — which mix to form other colors on the screen. Thus, as the reactions progress the color of the reaction products changes. Concentrations of all constituents, and any mathematical combinations of them, can be graphed in real time. *OOTLs* also models diffusion processes. Multiple reactors can be created and linked in linear or two-dimensional arrays. Diffusion constants can be specified, and the resulting dynamics displayed by means of animated colors. Since the diffusion constants of the different constituents need not be the same, the effects of variation in this important parameter are directly observable. *OOTLs* can function as a gateway to many different topics in various areas of science and mathematics.

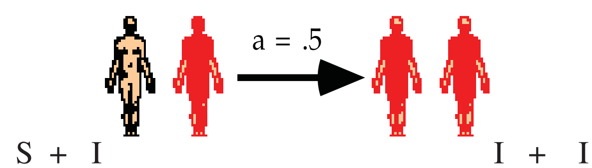
The following example illustrates the use of *OOTLs.* The application describes a classic situation in epidemiology: the spread of disease in a large population concentrated in a local geographic area. A familiar example is mononucleosis (the "kissing disease") spread among students living close to each other in university dormitories. The basic model assumes that most students will eventually contract the disease through contact with a student who is infected, and that each student who becomes infected will eventually recover and acquire immunity. Thus, there are three sub-populations of students at any time. There are the *Susceptible* students, those who have not yet caught mononucleosis but who will catch it if they come in contact with an infected student; the *Infected* students, those who are currently ill; and the Recovered students, those who have been ill and are now immune.

The system of ordinary differential equations (ODEs) describing this dynamic model involves three populations of individuals. It is defined as follows (where a is the transmission rate, the fraction of the individuals in the susceptible population that becomes infected per encounter per day; and b is the recovery rate, the fraction of the individuals in the infected population that recover per day):

(1) $dS/dt = -a * S * I$ =change in Susceptible
(2) $dI/dt = a * S * I - b * I$ =change in Infected
(3) $dR/dt = b * I$ =change in Recovered

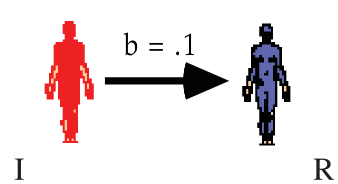
For each susceptible individual that gets ill, S is decreased by the same amount as I is increased, thus the term $a * S * I$ appears twice, once negative, once positive. The same applies to the recovery rate term, $b * I$, though it is offset by only one equation. Our experience, and that of other investigators, is that most high-school students are unable to formulate these rate equations.

This is how students might build the same spread of disease model using *OOTLs*. They begin by identifying the types of objects that are relevant. In this instance they identify two kinds of objects — individuals who are currently infected (denoted *I*), and those who are healthy but susceptible (denoted *S*). They then describe the possible interactions between such individuals that can give rise to the observed behaviors — transmitting or "catching" the disease. In this case, the students identify a single interaction: "When a susceptible individual meets an infected one, the healthy individual becomes infected also." They specify an interaction rate, a. They then define and select the icons specifying susceptible and infected individuals, and arrange them to form the following causal *OOTLs* interaction equation, describing what occurs before and after the two types of individuals come into contact.



Once this transformation equation has been input via the *OOTLs* graphical interface, students can simulate the system based on the initial conditions they choose. If they start with a small number of sick people and a large number of healthy ones, over time all the healthy individuals will "turn into" sick ones, reaching a stable final state, though the dynamics involved in attaining this are not trivial.

Students are then asked whether this is the actual outcome that describes what happens in the real world. Their considered answer is: "No, people do not stay sick forever. They get better." The issue they now address is: how do people stop being sick, and how is this to be represented? One way to extend their model is to simply allow for sick individuals to become healthy again after a period of time. This requires creating a new type of object (denoted *R*), for individuals that have recovered and are immune to further infection. Then, a second transformation equation is added to the model, expressing recovery: sick individuals eventually recover at some rate *b*.



This is known as a first-order decay, and produces exponential diminution over time. The result of simulations with this new two-equation model now yields a peak level of infection, with the number of infected dropping thereafter, followed by a new stable state in which not all the original healthy (susceptible) individuals may become sick. Students can extend the model by adding additional transformations of increasing complexity, such as the addition of a rule to allow recovered healthy players to again become susceptible to infection over time. Alternatively, recovered individuals could still be carriers without any outer symptoms, thereby infecting healthy individuals. And finally, students might incorporate population dynamics, allowing individuals to reproduce, die, and form sub-populations with different rates for growth and death.

In realizing these models, the appropriate mathematics is handled by the *OOTLs* graphics language preprocessor. Notice that, while the differential equations (DE) representation employs three equations, one for each possible health state of the individual, in *OOTLs* only two process equations are required — the DE form is redundant, and beginning students are often confused by the significance of its terms. The dynamics of the DEs are fully captured by *OOTLs*, as illustrated by the simulation output in Figure 13.27.

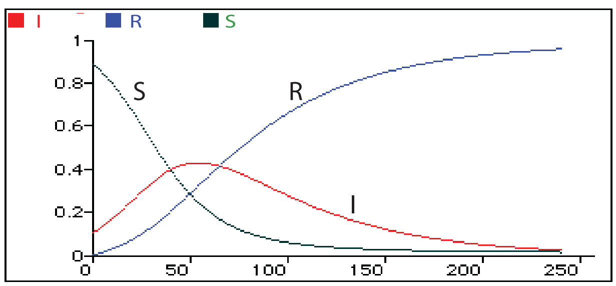


Figure 13.27 Simulation output.

The dynamics resulting from this formulation display the classic onset and course of an epidemic, with the number of infected peaking at a certain time, and then diminishing as the number of recovered increases asymptotically. Note however, that not all susceptible individuals will necessarily get ill. If the rate of spread is not as fast as the recovery, then some individuals escape infection. However, decreasing the rate of recovery (i.e., lengthening the incubation-illness period) has the effect of ensuring that more individuals will get the disease. This important concept is very easily explored in the process-specific form embodied in *OOTLs*. The *OOTLs* system provides its own DE simulation engine. However, *OOTLs* can also be used as a language front-end to drive other simulation engines, including those employing discrete and stochastic mechanisms as well as those employing continuous dynamics.

13.4 Learning and teaching language and reading

BBN research in educational technology has covered a wide range of issues related to language processing and comprehension. Applications have included teaching language and reading skills to beginning learners and to those with severe hearing impairments.

Second language learning

People who learn a second language as adults often speak it with a “foreign” accent all their lives, in spite of using it daily. One explanation for this is that in the course of learning one's native language, one loses the ability to make certain auditory discriminations or articulatory movements that are not characteristic of that language. Thus if the second language requires such discriminations or movements, one may not only have difficulty making them, but may be unaware of the fact. To one's own ear one's pronunciation sounds correct, even though to the ear of a native speaker of the second language it does not.

The question naturally arises as to whether distinctions that are difficult to make by ear might be more susceptible to training if they could be made by eye. A computer-based system was developed at BBN in order to explore this possibility. The system was built around a DEC PDP-8L computer equipped with a bank of analog filters to pre-process incoming speech, a tape recorder with a five-second loop to maintain a continuous recording of the last five seconds of speech input, and a cathode ray tube on which the results of various types of speech analysis could be displayed. The system, which was called the Automated Pronunciation Instructor (API), was used in a series at studies aimed at developing better procedures to teach the correct pronunciation of a second language. The second languages used in these studies were English for native speakers of Spanish and Mandarin Chinese for native speakers of English.

Several displays were developed, each emphasizing some particular aspect of pronunciation that was deemed by the investigators to be particularly relevant to the training objectives. One such display showed a schematized representation of tongue position during vowel production. Tongue position here means roughly the two-dimensional position of the tongue hump within the vocal track as viewed from the side. (The vowels in Spanish and English are not quite the same and native speakers of Spanish often substitute the nearest equivalent Spanish vowel for the correct English vowel when speaking English.)

Inasmuch as vowel quality is determined, in large part, by the position of the tongue body in the mouth, it was hypothesized that a display that permitted the student to compare the actual tongue position with the desired tongue position for specified vowels would facilitate correct pronunciation. Actual tongue position was represented by dots in a large rectangle on the display. The desired position (more accurately, the region of acceptable positions) was represented by a small rectangle within the large one. The student's task was to produce a vowel in such a way that the dots fell inside the small rectangle. Tongue position was inferred from certain sum and difference calculations performed on the outputs of the individual analog filters. Figure 13.28 shows the tongue position display as it might appear for both a correctly (on the left) and an incorrectly pronounced vowel (on the right). The word the student was attempting to pronounce was <u>bow</u>.

A second difficulty that native speakers of Spanish have in learning to speak English, is that of producing initial aspirated stops /p,t,,k /. A native speaker of English delays voicing onset following these initial stop consonants for a few tens of milliseconds. A common error made by a native speaker of Spanish is to initiate voicing too soon, thus making what should be unvoiced consonants sound like their voiced counterparts. Therefore, a program was also written to display aspiration time, shown in Figure 13.29.

The horizontal line at the bottom represents the segment of the utterance during which voicing occurred. The vertical line represents the point at which the stop was released. The dots that form a more-or-less parabolic curve represent aspiration intensity at successive ten millisecond intervals. The API system was used in a variety of experimental training situations (Kalikow and Swets, 1972).

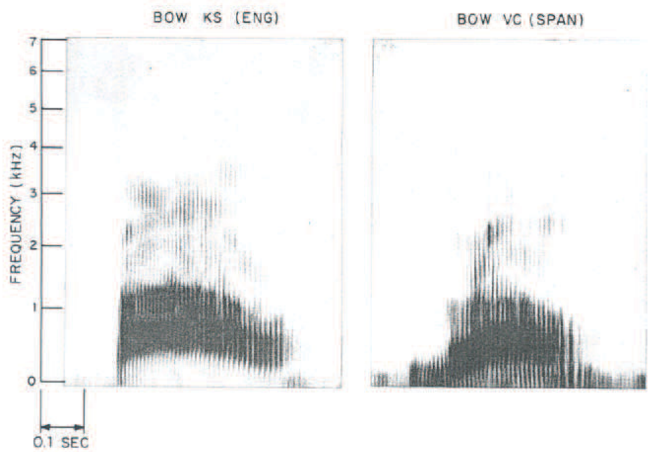


Figure 13.28 Tongue position displays.

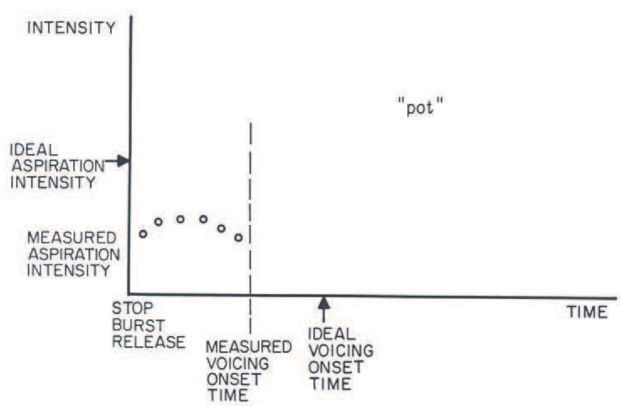


Figure 13.29 Aspiration time display.

Speech training aids for the deaf

An outgrowth of the work on second language learning was the development of an experimental computer-based system of visual displays that could be applied to the problem of improving the speech of pre-lingually deaf children. Children who are born deaf or who become deaf during the first couple of years of life do not develop normal speech capability, because they are deprived of the feedback channel through which hearing children learn to speak by comparing their utterances to those of other people around them. The inability to communicate via speech has profound implications for educational, social, and vocational development. The use of visual displays to help teach the correct pronunciation of a foreign language quite naturally led to the thought that such displays might also be useful in teaching speech to deaf children. The displays would not necessarily be the same, of course, but the basic idea of analyzing speech in a variety of ways, representing the results of those analyses visually, and providing students with visual targets to match seemed transferable to the new problem context. A system similar in many respects to the Automatic Pronunciation Instructor was designed and built (Nickerson and Stevens, 1973; Nickerson, Kalikow, and Stevens, 1976). The computer was a DEC PDP-8E. The configuration of peripherals was slightly different, but, like the earlier system, this one also contained a CRT display. Precisely what to display by way of speech properties was not clear a priori. It is not as though the speech of deaf children typically needs a little fixing. The problems tend to be numerous and complex (Nickerson, 1975; Nickerson and Stevens, 1980; Nickerson et al., 1983). They cannot be worked on all at once and there was very little in the literature to give guidance regarding where best to start. With the intent of providing a basis for exploration, the BBN system was programmed to produce several types of displays. Some of these were intended to support vocal exercises in game-like situations; others provided continuous feedback regarding one or more specific speech parameters during the emission of connected speech. The properties or speech that the system could display included amplitude, fundamental frequency, nasalization, and spectral distribution. One game-like display is illustrated in Figure 13.30. It shows a ball moving at a constant speed from left to right across the screen.

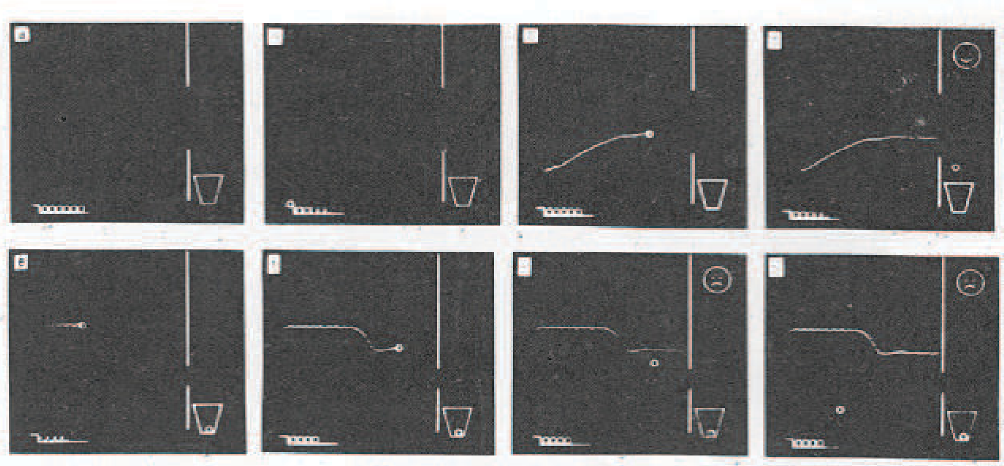


Figure 13.30 Pitch-controlled game display.

The height of the ball was controlled by the pitch of the speaker's voice. A vertical line positioned toward the right side of the screen represented a "wall" with a "hole" in it. The student's task was to adjust the pitch of his voice so that when the ball arrived at the wall it would be at precisely the right height to pass through the hole. If it did, it then dropped into a basket on the far side of the wall and a smiling face appeared in the upper right corner of the display. It the ball arrived above or below the hole, it rebounded to the left. Both the height and width of the hole could be adjusted by turning a control knob. The top sequence shows a successful trial; the bottom one an unsuccessful one.

In a more complicated version of the game, two walls were used, separated by an adjustable difference. The heights of the holes in these walls could be different, thus forcing the student to change the pitch of his voice during a short time period, in order to get the ball though both holes. This game was used to teach students to control the pitch of their voices and in particular, to drop the pitch at the end of an utterance, which is something hearing speakers spontaneously do, but prelingually deaf children typically do not.

The display used most often with students was one that showed speech amplitude as a function of time. This display was used in training sessions aimed at improving the temporal properties of the children's speech. The need for such training is illustrated by the fact that one characteristic of the speech of deaf children is a lack of differentiation between stressed and unstressed syllables. In the speech of hearing speakers, stressed syllables may be slightly louder than unstressed syllables, and almost invariably are considerably longer in duration. The amplitude-versus-time display was used to help the deaf children modify the temporal characteristics of their speech, bringing them more in line with the temporal patterns produced by hearing speakers. The usual approach was for the teacher to illustrate the appropriate timing of an utterance by making the utterance and displaying its temporal pattern on the top half of the display. This pattern would remain in view as the student attempted to produce one on the bottom half of the display that would approximately match it.

Two of these systems were built and installed in two schools for the deaf — the Clarke School for the Deaf in Northhampton Mass., and the Lexington School for the Deaf in New York City where they were used on an experimental basis for several years. Some formal experiments were done to determine whether training procedures based on the use of specific displays would be effective in modifying the speech of deaf children in desired ways, and in particular with respect to nasalization, fundamental frequency, timing, and voice quality. This work was documented in a series of reports (Nickerson and Stevens, 1980; Stevens, Nickerson, Boothroyd, and Rollins, 1976).

While the speech of most of the participating students was modified in ways targeted in their training objectives, measured improvements in intelligibility were not consistently realized. One general conclusion that came out of the project was that there is a need for greater knowledge of how the intelligibility of speech depends on its objectively measurable properties. It is relatively easy to specify various ways in which the speech of a particular deaf child differs from the norm. However, given our current state of knowledge, it is difficult to say which aspects of the speech one should attempt to change in the interest of affecting the greatest improvement in intelligibility during limited training time.

In addition to being used in formal experiments, the systems were also employed at the schools where they were installed for a variety of other purposes. These included making measurements on children's speech for purposes of diagnosis, self tutoring (some children used the systems on their own to help wok on specific aspects of their speech), and teacher training. Several additional efforts to apply computer technology to the problem of enhancing the speech of deal children have been initiated since the completion of this project, both in this country and elsewhere.

Reading Aide

The number of adults in the population with unacceptable levels of literacy is enormous. Illiteracy costs the United States over 225 billion dollars annually in corporate retraining, lost competitiveness, and industrial accidents. The implication is clear: our goal of providing a modern competitive workforce hinges very directly on our ability to achieve a massive improvement in adult functional literacy during the next decade. This cannot be accomplished through the use of human teaching alone. There simply are not enough reading instructors in the country. Their teaching must be augmented by the creation and widespread application of an effective technology for automating literacy tutoring. More than one out of five adult Americans is functionally illiterate and their ranks are swelling by about 2.3 million persons each year. Nearly 40 percent of minority youth and 30 percent of semiskilled and unskilled workers are illiterate.

Although for a small fraction of illiterates the ability to read is impeded due to neurological problems, and for others there are learning difficulties that are not associated with sensory or motor problems, the primary cause of illiteracy among Americans is a failure to learn to read. For most adult illiterates, a major obstacle to effective reading development lies in two simple facts. The human resources do not exist to provide the teaching support that is needed, and there is no way of adequately increasing their number to provide such support during the next several years. We cannot develop a sufficient force of trained professionals and paraprofessionals at the level of expertise required, even with a massive injection of funding. The only option we have is the effective introduction of appropriate technology.

Learning to read requires time and practice. Research indicates that once the basics of learning to read are in place, a grade-level gain in reading ability takes approximately 100 hours of engaged literacy training. Further, at beginning levels of reading, individual feedback, motivation, and guidance are critical. Studies show that students need 4-10 minutes each day of supported reading to progress for 1st to 2nd grade, and 20 minutes per day to progress from 3rd to 4th grade. In 1996, Marilyn Adams, Richard Schwartz, and Madelyn Bates developed a computer-based *Reading Aide* to address the early reading problem. It incorporates capabilities for advanced speech recognition and sophisticated speech analysis. It operates as follows. The computer displays a page from a book, indicates a sentence for the child to read, listens to the child reading, highlights words as they are read, detects dysfluencies, and responds accordingly. Students read at their own level and at their own rate.

The *Reading Aide* can detect a wide range of dysfluencies. Examples include incorrect words, skipped words, repeated words, stutters, starting over, getting stuck, hesitation, and mechanical reading. It responds appropriately — for example, by moving on, asking for a retry, reading to the student, and providing help when necessary. It can delegate some control of the system to the student. It has numerous modes, from a line at a time to "read me a story." The student can navigate within a story either forward or back, play a word or sentence, or request help. The intent is to maximize the detection of dysfluencies and to minimize false alarms. However, speech recognition is imperfect. Moreover, many early readers have immature elocution, some children have strong accents, and new readers' oral reading is not fluent. The system incorporates an explicit model of dysfluencies. The sentence grammar includes distinct probabilities for skipping, repeating, starting over, and getting stuck. The single word grammar includes distinct probabilities for arbitrary phoneme sequencing, stuttering, and likely substitutions. The system avoids making responses that are confusing. For example, telling a child he made a mistake when he didn't can be extremely damaging. Instead, its responses are designed to be informative (useful) but noncommittal. The system keeps a log of the confidence of its responses for later analysis. It maps the responses into a decision tree, annotating each sentence with the acceptability of *each* possible response. Figure 13.31 shows the system architecture (Gifford and Adams, 1996).

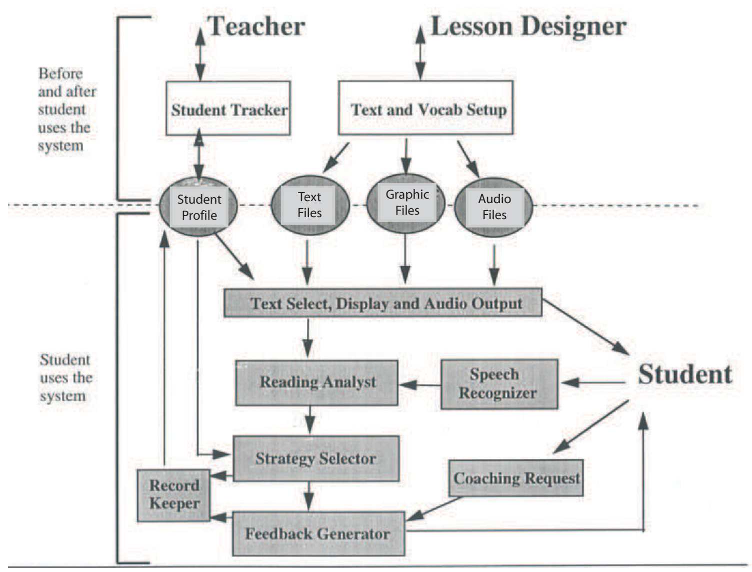


Figure 13.31 *Reading Aide* system architecture.

ILIAD

Dr. Lyn Bates, was Co-Principal Investigator with Dr. Kirk Wilson of Boston University on the project "Interactive Language Instruction Assistance for the Deaf," funded by the Bureau of Education for the Handicapped. The research was motivated by the fact that children who are prelingually deaf often never master standard English and usually lag far behind their grade level in reading English. Bates and Wilson hypothesized that one major reason is that they had not been exposed to many examples of certain key English structures, such as passive sentence forms. (An example: "The car was hit by the truck." as contrasted with the active form "The truck hit the car.") Readers need experiences with these kinds of English structures to understand how the syntactic structure affects the meaning of sentences.

To address this problem, they constructed *ILIAD*, a computer system that employed a transformational grammar to generate English sentences with random components. Particular features (such as passives, possessives, plurals, and various irregular forms) could be presented by settings under the control of the teacher. The system generated sentences instead of running through a fixed list, thus it never ran out of examples; students were not bored by having to repeat the same material over and over again.

ILIAD provided students with several game-like exercises. The system was used in classes at the Boston School for the Deaf (Bates and Wilson, 1981). The project won a Merit Award from Johns Hopkins University in the area of Personal Computing to Aid the Handicapped in 1981.

13.5 Training technology

From its early years BBN has been engaged in research and development involving the application of technology to technical training in complex task domains. Much of the work has focused on the introduction of new approaches employing sophisticated computer-based instructional technology based on methods derived from artificial intelligence and computational modeling. This section describes several such training applications in complex systems operation and maintenance tasks as well as aircraft piloting and tactical decision making tasks requiring support for real-time responses.

STEAMER

STEAMER was an advanced computer-based interactive graphics-oriented expert system for training operation and maintenance of complex steam propulsion power plants. It was developed by Bruce Roberts, Albert Stevens, Larry Stead, Albert Boulanger, and Glenn Abrett, under support of the Navy Personnel Research and Development Center in 1978–1983. A Navy steam propulsion plant is a very complex physical system consisting of thousands of components interconnected by miles of pipes, and requiring the operation of a team of 16 to 25 individuals. Years of instruction and experience are required to develop the understanding and skill for competent operation of a plant. The driving idea behind *STEAMER* was to enhance operator training through the development of a propulsion plant multi-level simulation with a color graphics interface and an intelligent tutoring component capable of instruction, guidance, and explanation of plant operation, operating procedures, and underlying operational principles (Stevens, Roberts, and Stead, 1983; Stevens et al., 1981; Hollan, Hutchins, and Weitzman, 1984).

Using an AI model of the propulsion plant, *STEAMER* generated interactive graphical diagrams of the entire plant and individual plant components at different levels of detail. The propulsion plant comprises many subsystems. *STEAMER* graphically depicted the flow of water or steam through these systems and the effects that various types of operator actions and system malfunctions would have on the operation of the plant. A screen shot of the *STEAMER* main steam cycle is shown at the left of Figure 13.32. An interactive diagram of the main engine lube system in *STEAMER* is shown on the right.

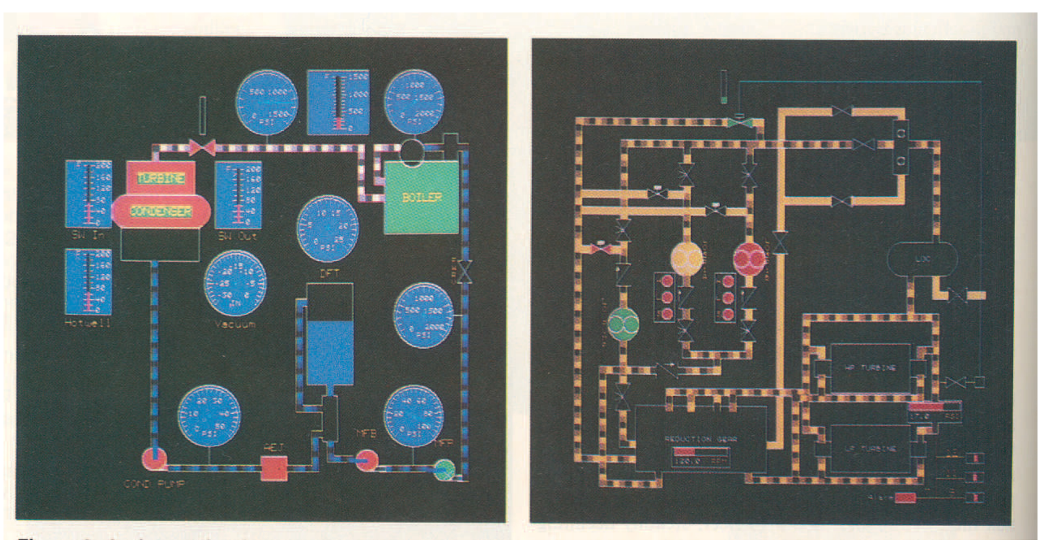


Figure 13.32 *STEAMER* screen shots.

The *STEAMER* instructional system provided a structured tutoring mode that presented problems to the student and guided him through a lesson. It supported exploratory learning activities that enabled students to perform "what if" experiments to discover the consequences of various operator actions. It could generate explanations of the operation of the plant, of what is happening, as the simulation is run. Thus, it could teach not only the plant's operating procedures, but also their underlying rationale. In describing a procedure for draining a chamber, it could explain the reason for the order of operations, e.g., why it is necessary to align the chamber's drain valves before opening an input valve to the chamber. (Because, otherwise, the water that is left in the chamber will mix with the steam, and high-energy water will get thrown downstream.)

STEAMER served as a compelling demonstration of the great potential of animated graphics representations driven by AI simulation models for making visually clear and understandable the dynamic interactive operation of complex physical systems comprising large-scale multi-level logical, electrical, and material components. A prototype *STEAMER* system was tested in a Navy training course. The system enabled students to inspect and operate a propulsion plant at various conceptual levels. Students found it easy to use, and programmers and curriculum developers found that its graphic editor readily enabled them to add and modify *STEAMER* diagrams. It was well received by users within the Navy training command.

ORLY

The *ORLY* flight training simulator was developed and employed in flight performance analysis research in 1974-1978 under support of the Naval Training Equipment Center and the Air Force Office of Scientific Research. The *ORLY* system development and instructional research was performed at BBN by computer scientists Wallace Feurzeig, George Lukas, Joe Berkovitz, Bill Huggins, Dan Stefanescu, Marty Schiff, and consultants Dan Cohen, Ken MacDonald, and Pat McHugh (Lukas, Feurzeig, and Cohen, 1975; Feurzeig, Lukas, and Stefanescu, 1980). The goal of the *ORLY* project was the development of computer-based methods for diagnostic performance analysis. The major research product was a performance analysis system for providing very specific characterizations of student pilots' performance on a variety of instrument flight control tasks. It enabled, not only the detection of performance errors, but also the generation of diagnoses characterizing the students' underlying difficulties. The first versions of the *ORLY* flight simulator were implemented on the DEC PDP-15 and PDP 1/PDP-10 computer systems by Cohen. The final version and the associated performance analysis facilities were implemented at BBN on a DEC GT-44 graphics display system with a PDP-11 CPU.

ORLY presented a realistic and fairly complete set of stylized but fairly realistic instruments on the bottom half of the display. The panel provided standard presentations of attitude, airspeed, altitude, heading, rate of climb, rate of turn, power, time, a compass rose/digital compass, an automatic direction finder (ADF), and an instrument landing system (ILS). The outer marker was at the ADF and a beeping and flashing of the corresponding instruments indicated passage over it and over the middle marker. A schematic and sparse cockpit window view occupied the top half of the display screen. The objects presented were the crossed airport runways, block structures corresponding to airport structures and antenna towers, and a graduated cloud ceiling. Distant mountains provided horizon information. The window view was primarily used for take off and landing. Figure 13.33 shows the *ORLY* instrument display and window view at three successive stages of a final approach.

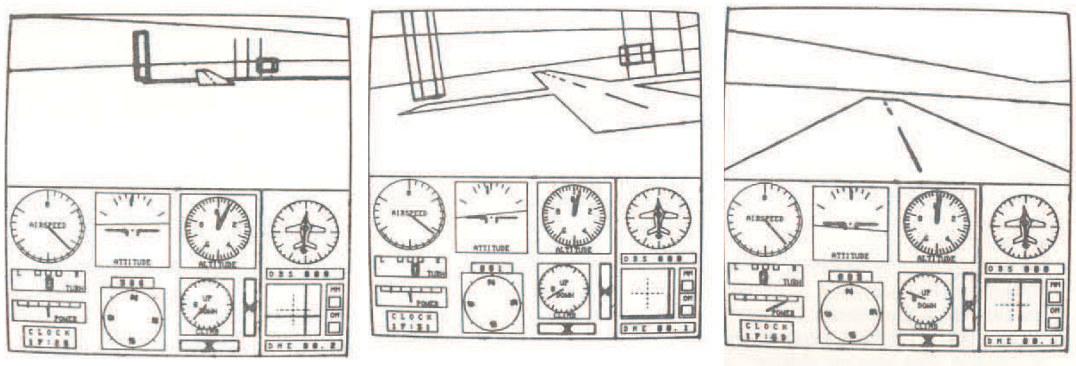


Figure 13.33 *ORLY* instrument display and window view.

The performance analysis system employed state-of-the-art computer feature extraction and pattern recognition methods expressly designed to mirror the analysis procedures used by expert instructors. In the first phase of the analysis, pilot performance data on task-dependent flight parameters (such as glide path, heading, rate of turn, etc. for instrument approaches) are fitted by a connected sequence of line segments. In the second phase, each segment is labeled by a set of attributes that characterize its performance relative to prescribed course and tolerance regions associated with that parameter in the flight plan. A segment is characterized by its location with respect to the tolerance region, by its length (duration), and by its slope. In the third phase, sequences of labeled segments, both within and across parameters, are interpreted as control patterns. Error and correction patterns of various types are identified. The control patterns include under-corrections, over-corrections, oscillations, and stabilizations. These patterns are specified in the program by formal procedures. In the last phase, this set of partial descriptions is integrated to produce an analysis narrative describing salient features of the pilot's performance during the task. Errors, error patterns, and contextual information to help explain difficulties are identified.

These methods were successfully applied to the analysis of basic instrument flight tasks, e.g., closed patterns such as figure 8s and cloverleafs, incorporating climbs, descents, timing variations, and airspeed changes. Figure 13.34 shows two such patterns.

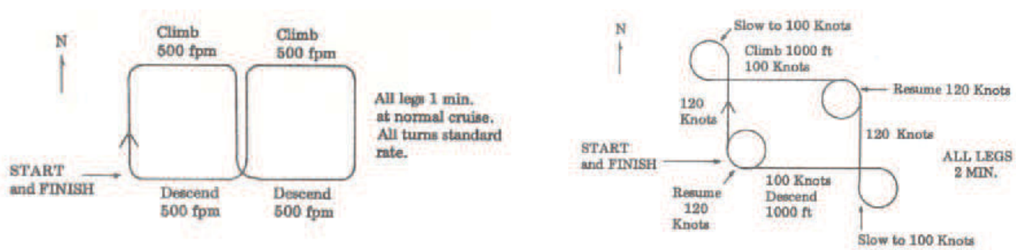


Figure 13.34 Basic flight patterns.

A typical flight map generated by a student pilot flight trial of the first pattern above and the chart record generated by *ORLY* for this flight are shown in Figure 13.35. The student's rate of climb, rate of turn, heading, altitude, airspeed, and power are charted

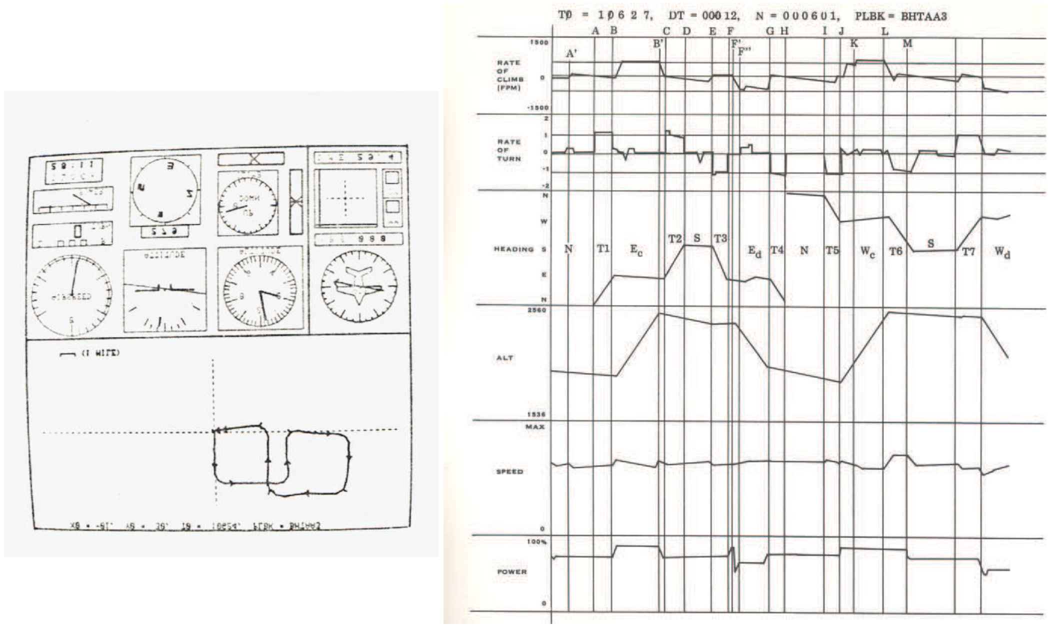


Figure 13.35 Typical flight map and chart record.

across all flight segments. A multi-level, context-sensitive, task-dependent procedure employing a pattern recognition grammar then performs a detailed analysis of the chart data. It recognizes standard performance patterns as well as canonical errors such as ballooning, diving, beginning a turn in the wrong direction, waiting too long to begin roll-out on a new heading, improper use of pitch and power, over-banking on turn entry, increase or decrease of bank during a turn, erratic bank control, shaky turn, beginning roll-out in the wrong direction, and climb/descent instead of descent/climb.

During the last phase of the analysis, the system produces a coherent summary of the significant features of the student's performance in language familiar to instructor pilots. The system was used in the analysis of 150 closed turn patterns flown by 16 student pilots. Comparisons of instructor analysis of these flights established that all unequivocal errors and error correction patterns were found and correctly identified by the system (Lukas, Berkovitz, and Feurzeig, 1977). For example, the system's description of the student's heading control during the first straight leg in the figure 8 trial shown above was: "The pilot established the heading and maintained the course for 53 seconds. An uncorrected drift then occurred while the pilot was having difficulty with altitude control." An instructor pilot's description of the same leg was: "Pilot drifts from heading until he is well outside the tolerance range. The pilot is apparently occupied by the altitude adjustments he has to make when the drift occurs." Instructors judged the performance summaries to be correct and essentially complete at that level of description.

Subsequent work with *ORLY* involved more complex types of maneuvers involving navigation components as well as instrument control. Among these tasks were holding patterns and ADF and ILS approaches, including missed approaches. These tasks entailed new and more complex errors involving, for example, glide slope and localizer parameters. In some phases of this work, pilots were given a great deal more latitude in their choice of flight path and in their mode of execution of the maneuvers. Thus, the unambiguous determination of certain errors was more difficult than for tasks with completely prescribed plans. Despite the increased task complexity, the methods used for analysis of turn pattern tasks also appeared effective in the analysis of navigation tasks.

TRIO

TRIO (Trainer for Radar Intercept Operations) is an expert instructional system for training F-14 interceptor pilots and radar officers in dynamic spatial reasoning and the basic tactics of high-speed air intercepts. It was developed by Wallace Feurzeig, Frank Ritter, William Ash, Barbara White, and Michael Harris under support from the Navy Training Systems Center in 1983-1988 (Feurzeig, Ash, and Ricard, 1984; Ritter and Feurzeig, 1988). *TRIO* was designed to provide training in the effective conduct of air intercept operations by an F-14 radar intercept officer (RIO) in defense of an aircraft carrier or other naval asset. The *TRIO* task environment supports simulations of airborne radars, interceptor and target aircraft operations, and weapons models. It provides dynamic displays of heading, bearing and displacement vectors, radar screens, flight instruments, intercept parameters, radar and missile envelopes, and interceptor/target aircraft ground tracks. It incorporates real-time speech recognition and synthesis subsystems including advanced capabilities for recognition of naturally articulated utterances from an extensive lexicon. *TRIO* supports three instructional modes: demonstrations by the *TRIO* expert program, student practice with optional guidance, and performance analysis and student debriefing following student practice (Panagos, Feurzeig, and Ritter, 1987).

TRIO was the first successful application of intelligent tutoring system technology to real-time tactical task domains. In the *TRIO* environment, trainees participate in simulated engagements under the guidance of expert software tools that incorporate knowledge of the task and the training issues. These programs can demonstrate correct intercept tactics, provide assistance to correct trainee misconceptions, evaluate trainee task performance, and adaptively generate reasoned explanations of effective strategies. During these engagements, trainees observe indicators of system function (including simulated sensor output) and manipulate standard aircraft system controls.

An expert program in *TRIO* is capable of performing the same intercept tasks that it trains. The *TRIO* expert is articulate — as it performs air intercept engagements it explains its performance along the way. Each time it takes an action (e.g., calls for a change in heading, altitude or airspeed, selects or fires a weapon, or changes the radar display presentation) it can state the reason for the action, not only what the action is intended to accomplish but also why this is desirable in terms of its current goal. The goal structures of the tactics employed in performing intercepts are explicitly represented in the rules that drive the *TRIO* expert. This enables rapid evaluation and execution of the rules and facilitates real-time intercept performance in rapidly changing air battle situations. It also aids in the generation of tactically-based explanations for the expert's actions, to better motivate the sense and purpose of the strategic thinking and spatial reasoning involved.

The articulate expert capability is central to *TRIO*'s capability for instructional demonstrations. In the *TRIO* demonstration mode, the expert program runs *TRIO* to perform an intercept in very much the same way the trainee is expected to perform it. The intercept problem is usually assigned by an instructor or generated by *TRIO*, but problems also may be posed to the expert by the trainee. The expert explains its actions and the underlying reasons for them in terms of its current subgoal structure. The knowledge is represented using a special form of production rule system — continuous running, interrupt-driven, goal-directed rules — to operate the articulate expert program.

The expert performs intercepts in real-time and explains its actions and reasoning along the way. It uses the identical information the student sees and drives the simulation through the same interface. The intent is to provide the trainee with concrete models that prepare him for his own attempt to do similar intercepts.

After a trainee has seen the articulate expert fly an intercept to demonstrate a new tactical procedure or the application of a familiar tactical procedure to a new situation, he typically tries to do it on his own using *TRIO*'s guided practice facility. His performance is monitored and recorded for subsequent analysis. The trainee may try to perform the intercept without help. Otherwise, *TRIO* is able to intervene throughout the run to provide specific guidance to aid his performance, such as advance warnings to help trainees notice and avert major errors that threaten the success of the intercept. Rapidly changing tactical situations such as those occurring in air battle engagements impose very intense attentional demands. So the guidance offered by *TRIO*, which may come when the trainee is very absorbed in the intercept task, must be communicated in a clear and non-intrusive manner without stopping or slowing the action or breaking the trainee's concentration and thought processes. In real-time tasks with high cognitive loads, guidance must be presented in a way that allows a trainee to maintain his attention on the tactical situation while noticing and assimilating instructional communications. This is accomplished in *TRIO* by the use of "demons."

A demon is a continuously active rapidly executing program that monitors the state of task-critical parameters to detect a specific event, such as the imminent loss of radar contact or missile threshold. Demons are used primarily to detect and report errors in time for correction by the trainee. If its event occurs, a demon takes two actions. It records the event on the history list for use in the post-flight performance analysis debriefing narrative and it alerts the trainee to the need to take timely corrective action. The alert is communicated as a short message in a demon display window, possibly accompanied by flashing or by alerting sounds generated by the speech output device.

During an intercept exercise, *TRIO* employs speech recognition capabilities to simulate the voice communications between the RIO and the simulated pilot. The *TRIO* speech recognition facility is capable of real-time recognition of naturally spoken English messages from a specified lexicon of allowable RIO utterances, including fairly complex flight directives such as "Come starboard hard as possible to a heading of two four zero degrees." The pilot carries out the flight directives spoken by the RIO trainee as he directs the intercept. The pilot's flight control operations and the interceptor's flight dynamics are simulated by programs. *TRIO* acknowledges each RIO directive and executes it exactly as a human pilot would (subject to limitations of flight dynamics and response time) by making appropriate changes in the instrument and radar displays.

At the conclusion of an exercise, *TRIO* replays the relevant segments of the intercept, reproducing the displays along the way, with an added ground track display that shows the effect of the trainee's actions, and comments on the trainee's performance at each action time. *TRIO* debriefs the trainee verbally, reviewing the development of the engagement, recalling trainee actions, evaluating the quality of trainee actions, indicating appropriate actions at each decision point, and providing a reasoned explanation in terms of established intercept doctrine to support the recommended action. The expert's actions and explanations are given as spoken utterances, using the real-time speech synthesis device. The use of speech as the output mode is necessary because the RIO trainee has to attend closely to the radar scope and tactical information displays that were generated during the rapidly changing air battle. This poses a heavy visual workload; the use of another modality that does not distract the trainee's monitoring of the displays or otherwise hamper his visual performance, is essential.

The analysis of the trainee's performance is based on the use of pattern matching methods that compare the trainee's actions to allowable performance behaviors defined by a solution state space. The solution space represents alternative solution paths during each phase of the intercept as permitted by the prescribed engagement rules and procedures. These paths allow considerable variation in the kind, number, and timing of trainee actions over those demonstrated by the expert program in its execution of an intercept. The analysis identifies faulty action sequences, i.e. those that could not be effective in realizing the appropriate subgoals in the intercept solution space, and determines very specific reasons for their unacceptability in terms of their adverse effects on the intercept. The analysis enables *TRIO* to generate explanations of what the trainee did wrong, where it happened, why it was wrong, and what he should have done instead.

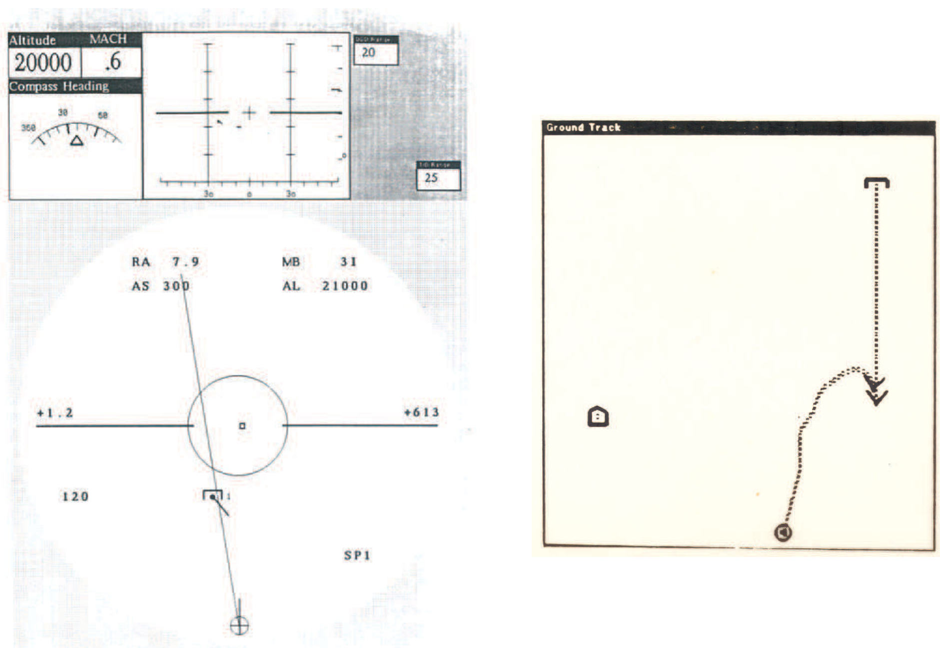


Figure 13.36 *TRIO* screen.

The *TRIO* screen is divided into a number of windows, as shown on the left side of Figure 13.36. These include displays of the RIO instruments (e.g., altitude, airspeed, heading, raw radar, and applicable target information). A text window provides the articulate expert's comments to the trainee when these are too verbose to present via the speech channel. The intercept track window is displayed during the debriefing mode following an intercept run. It shows the ground tracks of the RIO and the target aircraft that were generated during the run, as illustrated on the right side of the figure, which shows a successful intercept.

TRIO-based ideas, methods, and technology were incorporated in the BBN *INCOFT* project, described below. An operational version of the TRIO system was developed for training naval personnel at the Radar Intercept School in Pensacola, Florida.

MACH-III

MACH-III was a maintenance training aid computer for the Hawk system — an Intelligent Maintenance Trainer. It was developed in 1985-1989 by Dan Massey, Laura Kurland, Rob Granville, Dawn McLaughlin, Steve McDonald, Yvette Tenney, and Bruce Roberts.

(Massey, et al., 1986; Massey, deBruin, and Roberts, 1988; Tenney and Kurland, 1988; Kurland, Granville, and MacLaughlin, 1992). The work was done under contract to the U.S. Army Research Institute for the Behavioral and Social Sciences (ARI). BBN conducted an extensive series of experiments at the U.S. Army Air Defense Artillery School (AADASCH), Ft. Bliss, TX, to document the cognitive processes involved in successful (and unsuccessful) organizational maintenance of a complex electronic system, the AN/MPQ-57 High Powered Illuminating Radar (HIPIR) of the HAWK air defense system.

The understanding documented through these experiments was incorporated into the *MACH-III* intelligent interactive training system, which employed explanation-based reasoning to tutor trainee radar mechanics interactively in diagnosing and repairing faults in a simulated radar system. Acquiring expertise was a nontrivial task — the radar system comprised a number of subsystems with complex feedback loops, such as shown in Figure 13.37.

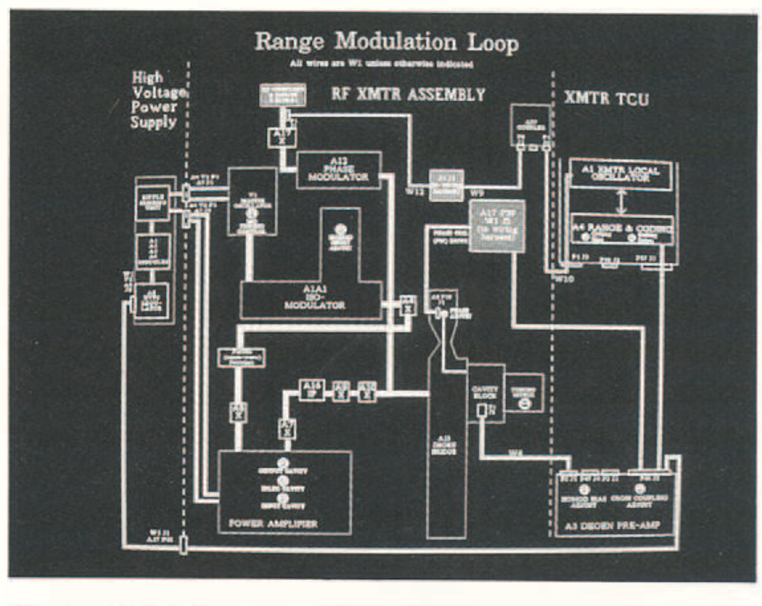


Figure 13.37 *MACH-III* subsystems.

The *MACH-III* system employed a novel approach to qualitative simulation of technical details of internal system functions and malfunctions. Using *MACH-III*, trainees explored the faulty behavior of a simulated system under the tutelage of task and tutorial expert programs. These programs demonstrated correct troubleshooting strategy, provided assistance to correct trainee misconceptions, prompted the recall of relevant knowledge, evaluated trainee performance (in the context of overall instructional goals), and adaptively generated reasoned explanations of system function and proper maintenance strategy.

In the qualitative simulation mode, the system enabled the trainee to manipulate all system controls and to observe all indicators of system function or malfunction. The system generated animated displays of the functional and physical organization of the radar, to help the trainee progress from understanding basic concepts to understanding the operation of the entire system. *MACH-III* included powerful facilities for adaptively generating reasoned explanations of system function and troubleshooting problem-solving strategies. The system was designed for use in conjunction with standard Government troubleshooting manuals.

MACH-III represented a significant new approach to training organizational maintenance personnel. The system simulation facilities of *MACH-III* made it possible to give the trainee cognitive experiences similar to troubleshooting a real system. Instead of the traditional focus on masses of seemingly unrelated mechanical details and procedures, the powerful simulation models in *MACH-III* gave students both the experience and the conceptual understanding that more closely characterized experts who had years of field experience. Thus, the trainee's time with *MACH-III* was efficiently spent on developing the reasoning skills essential to expert troubleshooting performance. Although *MACH-III* was a prototype system, and was not formally approved for instructional purposes, USAADASCH adopted it as an informal instructional device, supplementing their established training program of lectures and hands-on practice.

INCOFT

INCOFT (the Intelligent Conduct of Fire Trainer) was a training system for operators of the Patriot Air Defense system. It was developed in 1986-1989 by Dan Massey, Denis Newman, Wallace Feurzeig, Mario Grignetti, and Mark Gross (Newman, 1991) under contract to the Army Research Institute for Behavioral and Social Sciences (ARI), with sponsorship by the Joint Services Manpower and Training Technology Development Program of the Assistant Undersecretary of Defense for Life and Environmental Sciences. BBN developed *INCOFT* to teach the skills required for making real-time tactical decisions in complex air defense operations. *INCOFT* was designed for USAADASCH, Ft. Bliss, TX, to train Patriot Air Defense Tactical Control Officers (TCOs) and Tactical Control Assistants (TCAs).

A knowledge-based expert system, *INCOFT* demonstrated and explained basic concepts, provided individualized practice time, and evaluated performance. The system prepared trainees for higher performance in initial job assignments in 30 percent to 50 percent less time than existing non-adaptive simulators, which lacked tutorial capabilities. *INCOFT* faithfully reproduced the physical, functional, and tactical conditions related to the specific skills being taught. It provided a trainee workstation that closely replicated the appearance and functionality of Patriot operator workstations. *INCOFT* simulation software mimicked the functionality of TCO and TCA workstations in a realistic engagement, replicating system behavior with sufficient fidelity to support required observations and actions. Trainee actions were monitored in real time during scenario execution, with continuing classification of performance. Immediate feedback and after-action analysis reviews were provided via a speech synthesizer controlled by the intelligent training software.

The *INCOFT* system provided trainees with an easy-to-use interface and an interactive learning environment. It incorporated much of the system architecture, AI methods, instructional strategies, and simulation and communications programs of TRIO, the BBN Trainer for Radar Intercept Operators (Massey, Feurzeig, Downes-Martin, and Ritter, 1985). The system provided multiple instructional modes. Typically, critical operator errors resulted in immediate intervention by the training expert, while less critical errors and omissions were noted during scenario replay for after-action review. Trainees could use the system without constant tutoring by instructors during practice sessions. Instructors could focus their time and energy on more advanced and complex training issues. This resulted in intensified instruction, accelerated learning, improved performance in initial job assignments, and greater operational readiness (Newman, Grignetti, Gross, and Massey, 1992).

QUEST

QUEST (Qualitative Understanding of Electrical System Troubleshooting) was an Intelligent Computer-Aided Instruction system for teaching electrical system troubleshooting (Feurzeig et al, 1983; Feurzeig, 1985; Frederiksen and White, 1984, 1989; White and Frederiksen, 1985, 1986, 1990). *QUEST* used qualitative simulation methods to teach knowledge-based reasoning about circuit behavior and troubleshooting. Humans think about the behavior of phenomena and systems in a qualitatively different way from that used to describe such behavior in mathematical simulation models. Experts in a domain (not only beginning students) use qualitative modes of thought and qualitative models to reason about system behavior. Thus, though it is necessary to employ mathematical simulations to obtain precise detailed descriptions of system behaviors, we also sought to teach conceptually sound qualitative reasoning. The use of qualitative simulation models is valuable for producing understandable explanations and for generating animated displays to show dynamic behavior. This facilitates learning by fostering the student's development of effective mental models for understanding and reasoning about system behavior.

The *QUEST* expert system employed a qualitative simulation model for reasoning about the behavior of RLC electrical circuits composed of batteries, wires, resistors, coils, condensers, lamps, switches, and test lights (Ritter, 1986). *QUEST* was capable of modeling the dynamic behavior of capacitors and inductors in relatively complex circuits. The qualitative simulation included a description of the circuit topology, a runnable functional model for each device in the circuit, rules for evaluating device states at each time increment, and circuit tracing procedures to aid in evaluating conditions for device states. The program generated graphical representations of circuit operation. It was designed to support a dynamic presentation environment within which an expert troubleshooting program could demonstrate troubleshooting concepts and strategy. The expert tutor could be called to solve problems and to demonstrate to students the reasoning involved. *QUEST* also provided an instructional mode for supporting student practice on troubleshooting problems. The program generated explanations of circuit operation in both working and faulted states, employing the same qualitative reasoning principles used in the execution of the expert troubleshooting strategy.

The *QUEST* instructional system provided students with a problem-solving environment within which circuits could be built, tested, and modified. Some circuit problems challenged students to make predictions about circuit behavior or to troubleshoot circuit faults. The qualitative causal simulation was run to illustrate principles of reasoning about circuits. The expert troubleshooter operated in interaction with the simulation program as it demonstrated a strategy for isolating faults. It incorporated the same type of reasoning as that involved in predicting circuit behavior. When solving problems, students could call upon these programs to explain reasoning about circuit operation or troubleshooting logic. Each tutorial program utilized a model that expressed its reasoning at a level of explanation appropriate for that particular stage of instruction. The circuit simulation program explained the operation of circuits in faulted as well as working condition. The troubleshooting expert generated explanations of troubleshooting logic. The *QUEST* project was supported jointly by the U.S. Office of Naval Research and the U.S. Army Research Institute.

QUIMON

During the last months of the *QUEST* project, Feurzeig and Ritter designed and implemented the *QUEST* Instructional Monitor, *QUIMON*, which embodied a novel approach to cognitive analysis (Feurzeig and Ritter, 1988). The distinctive diagnostic feature of *QUIMON* that set it apart from other ICAI systems was the incorporation of the strategy of eliciting explicit information from the student about his troubleshooting actions throughout the course of the problem interaction. The student states what he hopes to learn from each of his actions on the circuit prior to its execution by the circuit simulator. After its execution he lists any conclusions about circuit faults he draws from seeing the simulator's effect on the state of the circuit. This strategic procedure engages the student as an active participant in facilitating the critique of his own problem work; he contributes valuable primary source information to aid the tutor in making more informed and more valid diagnostic inferences about the student's knowledge, bugs, and learning difficulties. This contrasts with the AI inferencing strategy of attempting to develop a cognitive model based on the student's actions without direct input from the student on the intent of his actions or the implications of their effects.

Here is a brief summary of the application of this strategy in a troubleshooting problem scenario. The student is presented with a (presumably faulty) circuit at a level of complexity appropriate to his current phase of training. As he acquires knowledge of the circuit's behavior, he is asked to develop and maintain a list of suspected faults. Initially, all circuit components may be suspect; at the end of an investigation the list will be reduced to those the student has isolated as faulty. The student investigates circuit behavior through a sequence of actions (e.g., flipping a switch, inserting a test probe, replacing a component). Before each action he is asked what he hopes to learn from performing it. He responds by selecting an item on the pre-action menu. After he responds, he calls the simulation to run. The simulation engine then carries out the requested action, and the student sees the effect of his action on the circuit behavior and state. He is asked what he has learned as a result of performing the requested action. He responds by selecting an item on the post-action menu. Following this response, the three-step process is repeated, continuing with the student's next troubleshooting action. This procedure generates a rich body of diagnostic data for the tutor. It also helps the student structure his approach to problem solving and develop more deliberate and reflective habits of thinking.

The interface is straightforward. The student answers a question by using a mouse to choose a response from a set of responses on the display. Possible student responses to the question "Why do you want to take this action?" on the pre-action menu include the following items: Don't know; To explore general circuit behavior; To test a component; To test the feed to a component; To test the ground side of a component; To replace a component. The student designates the component, feed, or ground of interest by pointing with the mouse. The circuit simulator then performs the requested action and changes the state of the circuit as appropriate.

After the requested action is taken by the circuit simulator, the program asks the student "What did you learn?" The post-action menu includes as possible student responses: Don't know; Identified component that may be faulty; Ruled out component as possible fault; Identified suspect subcircuit that may have a faulty component; Ruled out subcircuit as having a faulty component. Some answers require more than one response, e.g. the first might indicate that the student wants to add an entry to his list of possible faults, the second to point to the component or lasso the subcircuit suspected of being faulty, the third to designate the type of fault.

The elicitation procedure is designed to be non-intrusive and unforced. The student is advised that he does not have to be absolutely certain about the reason for every action he takes along the way to developing hypotheses. The direct manipulation point and click operation allows rapid and easy interaction. The session produces a substantial knowledge base of the student's plans and goals with minimal interference to his troubleshooting activity. This fine-grained information about the student's intentions, expectations and conclusions can be valuable for understanding his performance and making plausible diagnoses of his misconceptions and difficulties. Moreover, such information can only be elicited from the student: it is, at the very least, extremely difficult for an ICAI system based on present AI methods to infer the student's mental states from his surface behaviors. Thus, we believe that the *QUIMON* work provides an effective starting point for development of more competent student diagnostic models.

The addition of information about the student's intentions, expectations, and plans, as well as his observed actions, is essential to making informed and insightful diagnostic hypotheses. This approach to diagnosis integrates commonsense principles from cognitive science with AI inferencing methods. It enhances the power and reliability of ICAI inferencing. It enables a wide range of applications to complex maintenance and troubleshooting training. The approach has obvious limitations. It assumes the principle of rationality, that problem solving behavior, whether correct or not, is always rational even when based on incorrect knowledge. Further, the elicitation procedure is not applicable in situations where students lack the knowledge or the appropriate vocabulary and language for talking about their problem solving plans and goals. Also, in real-time situations, where tasks have to be performed "on the fly," there is little time available to discuss the student's actions along the way non-distractively. Other instructional methods are required here, like those illustrated in the work on *TRIO* (Ritter and Feurzeig, 1988).

13.6 Educational networking

Soon after the development of computer time-sharing, BBN researchers explored the use of this new communication technology in schools with educational projects such as Stringcomp, which enabled multiple users remote access to interactive computing facilities. Following the development of the ARPANET, BBN began to investigate the application of computer networking technology to provide new ways for people to work together to improve learning and teaching. This section describes representative projects that addressed these new opportunities and challenges.

Co-NECT

In 1992, BBN successfully competed in a national competition sponsored by the New American Schools Development Corporation (NASDC) to design a new generation of "break-the-mold" schools. BBN's winning design, called *Co-NECT*, was selected along with ten others from a field of nearly 700 proposals. *Co-NECT* provided a framework for school-wide reform combining successful teaching practices with a new kind of school organization — all supported by internetworking technologies. The Co-NECT schools project was directed by Bruce Goldberg, Henry Olds, and John Richards.

Co-NECT schools are organized around small clusters of students taught by a cross-disciplinary teaching team. Most students stay in the same cluster, with the same teachers, for at least two years. Working within national, state, and district guidelines, teachers used performance standards defining what graduating students should know and be able to do. The curriculum revolved around "authentic" interdisciplinary projects designed to give students an opportunity to acquire critical skills and understanding.

Faculty representatives of each cluster served on a school design team. Led by the building principal, with input from parents and other members of the community, the team set overall goals and monitored results. A sophisticated communications infrastructure gave Internet access to everyone in the school community. Every *Co-NECT* school and school district received individual attention from a support team headed by field representatives, consultation with members of the *Co-NECT* design team on an "as needed" basis, and involvement in teleconferences. Schools also took part in the *Co-NECT* Exchange (an Internet-based information service, electronic forum, and support tool), and *Co-NECT* Critical Friends (a program of reciprocal school visits). As a school developed its own internal capacity for sustained educational restructuring and growth, assistance from BBN continued, with increasing reliance on video conferencing and other means of remote support.

The long-range *Co-NECT* technology plan included communications technologies providing students and teachers access to people and information resources both inside and outside the school; ubiquitous access to networked computing tools to provide a solid support structure for project-oriented workgroups; a technology-enriched base of information sources, including video and audio as well as electronically accessible text, data, and graphics; powerful software tools to support many subject matter and skill-building elements of the curriculum; multimedia tools for both exploration and "publication"; networked software used to help manage the scheduling and project development needs of the clusters; and software tools for managing the certificate-based assessment system and for organizing and presenting student portfolios of selected work.

After two years of testing and refinement, the *Co-NECT* design was used by an expanding network of schools and districts around the country, including schools in Juneau, Alaska; Worcester, Massachusetts; Cincinnati, Ohio; Memphis, Tennessee; and Miami, Florida. *Co-NECT* left BBN in 1998 to operate as an independent organization centered in Cambridge, Massachusetts (Morrison, 1997).

National School Network Testbed

With the rapid growth of Internet use in the United States, it became increasingly important to understand how to use these new communication channels and resources most effectively in education. It takes considerable investment to create the technical and organizational infrastructure necessary to support wide participation across a community. There was a need to research and share information about successful models, the benefits as perceived by learning communities, and the investment required. An empirical base of knowledge was needed in order to make sound policy decisions about investment on the part of local, state, and national governments. The National School Network Testbed (*NSNT*) was funded by the National Science Foundation to help develop that knowledge.

Approximately 250 institutions participated in the Testbed, including over 150 individual schools. Phase I of the *NSNT*, conducted over 18 months from 1992 to the spring of 1994, resulted in an understanding of ways schools and other educational institutions could take advantage of internetworking to build their own local information infrastructure in support of desired reforms in education. BBN scientists Beverly Hunter and Denis Newman directed the BBN effort (Newman, 1993; Hunter, 1995).

Phase 2 of the project, which began in the fall of 1994 and continued through 1997, was designed to build models for developing relationships among schools and their communities through the use of telecommunication, so as to enrich the education of both children and adults. The project conducted a longitudinal study to investigate the effect of Internet technology on participating schools over the three-year period. The study collected descriptive and analytic data to determine the extent and nature of changes.

The report identified the need for a product that schools and other organizations might use to manage their own Internet services. In response to this need, BBN developed the BBN Internet Server, and made it available to schools and other educational organizations. The Data Communications journal gave it an award as Product of the Year in January 1995. Teachers, students, and administrators used the server for communication and data access, both within their organizations and throughout the international Internet. Most significant for educational settings is the fact that users could manage the day-to-day operation of the server without having to use its native environment (UNIX). Instead, a teacher could use an associated educational management program, FrontDoor, that communicated with the server to carry out tasks such as adding new users to group or individual accounts, creating or modifying electronic mailing lists and newsgroups, and publishing Web pages.

MuseNet

The Multi-User Simulation Environment Network project, *MuseNet*, was supported by the National Science Foundation in 1995 under the program "Networking Infrastructure for Education." The research was performed by Wallace Feurzeig, Paul Horwitz, Barry Kort, David Fagan, Kenneth Schroder, and Natasha Cherniak. The goal of the project was the development of scalable distributed networking technology for supporting collaborative environments for science and mathematics education. The project was designed to demonstrate the power of distributed server technology in addressing a key "scale up" problem posed by the dramatic growth of educational traffic on wide area networks. By distributing large, computationally intensive educational applications among multiple heterogeneous servers, *MuseNet* showed how to make considerably more efficient utilization of network resources with consequent improvements in client service and response times (BBN Systems and Technologies, 1996).

MuseNet was both a technology infrastructure demonstration project and an education research project. *MuseNet* sought to make a significant educational contribution by enabling real-time collaboration among users of science simulations and other computationally-intensive applications across the Internet. This called for the development and demonstration both of educational infrastructure for supporting collaborative interactions as well as technology infrastructure for supporting distributed educational applications.

The technology demonstration was built on prior work with the BBN Cronus distributed operational environment. The Cronus system was used to distribute and manage the operation of large-scale applications among multiple servers to sustain smooth operations and optimize response times. *MuseNet* employed the methods of the Cronus-distributed system to optimize server utilization across a large set of Muses dedicated to computationally intensive educational activities.

The participating Muses and their hosts included MariMUSE at Phoenix College in Arizona, MicroMuse at MIT, EcoMuse at theUniversity of Vermont, Bridge Muse at the University of Southern Maine, De Anza Muse at De Anza Community College in California, Graham and Parks Muse at the Graham and Parks public school in Cambridge, CyberMush at CNIDR, and two Muses at BBN, Academy Muse and WindsMare. There was enormous variation in the server load at each of these Muses at different times. Sometimes the level of user activity was extremely high at one site while it was fairly low at another. Further, there was a great difference in user loads and response times at any given site at different times. The disparity in server load within this multi-server community perfectly typified the general problem confronting wide area network management in the era of enormous and rapidly fluctuating network traffic. We showed how the use of *MuseNet* alleviated this problem through dynamic load balancing.

The networking infrastructure in *MuseNet* supported multi-user collaboration in science simulation and modeling activities by making modeling tools and applications operable within a MUSE environment. This new educational infrastructure combined two learning technologies that until then had been separate and unrelated. Work on computer simulation and modeling had been directed at fostering students' development of the "habits of mind" associated with scientific exploration and inquiry. Work on MUSEs had been directed at self-discovery and empowerment through shared encounters in text-based worlds constructed by students. We merged the two technologies by developing *MuseNet* facilities to support student communication and collaboration centered on the use of modeling and simulation tools and applications. Like current MUSEs, the use of this environment enabled students to meet over the net with each other and with teachers and scientists, on virtual field trips to diverse science modeling microworlds. It provided students with powerful facilities for supporting real-time collaboration on joint investigations employing modeling tools and applications.

This development made possible the integration of MUSEs with educational software tools and applications that were designed independently of MUSEs — programs like *GenScope*, the Geometer's Sketchpad, Function Machines, *RelLab*, Interactive Physics, Explorer Science, and other powerful modeling and simulation environments, particularly those that naturally lent themselves to collaborative activities. Students could thus work together, exploring, investigating, building, and modifying computational science structures and processes, using the social and conversational features of MUSEs to discuss their progress and to negotiate their moves in the course of running the programs.

To realize the integration, the following strategy was adopted. A datastream channel and associated multimedia channels on the server control, coordinate, and synchronize the commands for running the software as these commands are decided upon by the users through conversational negotiation on the MUSE. This mode of operation does not require high bandwidth networking — each time the model is run the only data that are transmitted are the commands for changing the state of the program and its current outputs. The integrated *MuseNet* system was demonstrated with two science simulation programs: Space (a 3-D astrophysical simulation of space travel), and *RelLab*, the BBN software for supporting students work in relativity experiments. These demonstrations showed the feasibility and educational benefits of integrating Muses and science simulations, through adding a social dimension to collaborative inquiry activities.

References

Abelson, H., and diSessa, A., *Turtle Geometry. The Computer as a Medium for ExploringMathematics.* MIT Press, Cambridge, MA.,1983.

Barowy, B., Richards, J., and Levin, D., Computer Simulations in the Science Classroom. *Journal of Science Education and Technology*, Vol. 1 No. 1, 1992.

Bates, M. and Wilson, K., ILIAD Project Final Report, BBN Report No.4771, BBN, Cambridge. MA. September, 1981.

BBN Systems and Technologies, MuseNet: A Scalable Architecture for Distributed Educational Applications, Final Report, April, 1996.

Brown, J. S., and Burton, R. R., Multiple Representations of Knowledge for Tutorial Reasoning. In Bobrow and Collins (Eds.) *Representation and Understanding: Studies in Cognitive Science.* New York, Academic Press, 1975.

Brown, J. S., Burton, R. R., and Bell A. G., SOPHIE: A Step Toward Creating a Reactive Learning

Environment. *International Journal of Man-Machine Studies, 7* (5), pp. 67–696, 1975.
Brown, J. S. and Burton, R. R., A Paradigmatic Example of an Artificially Intelligent Instructional System. *International Journal of Man-Machine Studies*, Vol. 10, 1978a.

Brown, J. S. and Burton, R. R., Diagnostic Models for Procedural Bugs in Basic Mathematical Skills. *Cognitive Science*, Vol. 2, pp. 155–192, 1978b.

Brown, J. S., Burton, R. R., and Hausmann, C., Representing and Using Procedural Bugs for Educational Purposes. *Proceedings, National Association for Computing Machinery*, October 1977.

Burton, R. R. and Brown, J. S., A Tutoring and Student Modelling Paradigm for Gaming Environments. *Proceedings of the ACM SIGCSE-SIGCUE Technical Symposium on Computer Science and Education*, February 1976.

Burton, R. R. and Brown, J. S., Toward a Natural Language Capability for Computer-Aided Instruction. *Readings in Natural Language Processing*, pp. 605–625,Morgan Kaufmann, 1979.

Burton, R. R. and Brown, J. S., An Investigation of Computer Coaching for Informal Learning Activities. *Intelligent Tutoring Systems*, pp. 79–98, Academic Press, 1982.

Carbonell, J. R., AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction. *IEEE Transactions on Man-Machine Systems 11* (4): 190–202, 1970.

Carbonell, J. R. and Collins, A. M., Natural Semantics in Artificial Intelligence. In *Proceedings, Third International Joint Conference on Artificial Intelligence*, 344–351, Stanford, Calif., 1973.

Carbonell, J. R. and Collins, A. M., and Warnock, E. H., Semantic inferential processing by computer. In J. Rose (Ed.), *Advances in cybernetics and systems.* London: Gordon & Breach, 1975.

Clancey, W. J., and Shortliffe, E. H., *Readings in Medical Artificial Intelligence: The First Decade.* Addison-Wesley Longman Publishing Co., Inc.. Boston, MA, 1984.

Clancey, W. J., Tutoring Rules for Guiding a Case Method Dialog. In Sleeman and Brown, (Eds.) *Intelligent Tutoring Systems*, Academic Press, London, 1982.

Collins, A. M., and Stevens, A.. Multiple Models of a Complex System. In (Snow, Frederico and Montague (Eds.) *Aptitude Learning and Instruction: Cognitive Processing Analysis.* Hillsdale, N.J., Erlbaum, 1980.

Collins, A. M., Adams M. J., and Pew, R. W., The Effectiveness of an Interactive Map Display in Tutoring Geography. *Journal of Educational Psychology*, 1978, 70, 1–7, 1978.

Collins, A. M., Warnock, E. H., and Passafiume, J. J., Analysis and Synthesis of Tutorial Dialogues. In: Bower (ed.) *The Psychology of Learning and Motivation: Advances in Research and Theory*, pp.49–87, Academic Press, New York. 1975.

Coulson, J. E., Programmed Learning and Computer-Based Instruction. Proceedings of the Conference on Application of Digital Computers to Automated Instruction. John Wiley & Sons, Inc., New York: 1962.

Cuoco, A., *Investigations in Algebra: Combinatorics and Number Theory.* MIT Press, Cambridge MA, 1990.

Cuoco, A., Action to Process: Constructing Functions from Algebra Word Problems. *Intelligent TutoringMedia*, Vol.4, No. 3/4, Learned Information Ltd., Oxford, U.K., November 1993.

Feurzeig, W., The Logo Lineage, book chapter in *Digital Deli*, Workman Press, 1984.

Feurzeig, W., Algebra as Logo. *Proceedings, MIT Logo'85 Conference*, MIT, Cambridge, MA, 1985.

Feurzeig, W., Algebra Slaves and Agents in a Logo-Based Mathematics Curriculum. Book chapter in *Artificial Intelligence and Education Volume One*, Ablex Publishing, Norwood, N.J., 1987

Feurzeig. W., Explaining Function Machines. *Intelligent Tutoring Media*, Vol. 4 Nos. 3,4, Learning Information Ltd., Oxford, U.K., 1993.

Feurzeig, W., Visualization in Educational Computer Modeling. Book chapter in *The Use of Computer Models for Explication, Analysis, and Experiential Learning*, Towne, de Jong, and Spada (Eds.), Springer-Verlag, 1994a.

Feurzeig, W., Visualization Tools for Model-Based Inquiry. Book chapter in Baker and O'Neil (Eds.) *Technology Assessment in Software Applications*. Lawrence Erlbaum, Hillsdale, N.J., 1994b.

Feurzeig, W., Function Machines Pre-Algebra Sequence, *CAETI Newsletter*, Number 14, `http://isx.com:80/CAETI/newsletters/newsletter.14.html`, 1997.

Feurzeig, W. A Visual Language for Model-Based Mathematics Inquiry, book chapter in *Modeling and Simulation in Science and Mathematics Education*, W. Feurzeig and N. Roberts (Eds.), Springer-Verlag, New York, N.Y. 1999.

Feurzeig, W., Ash, W., and Ricard, G., The Use of Artificial Intelligence Methods in Real-Time Tactical Training. *Proceedings, American Education Research Association*, April 1984.

Feurzeig, W., Horwitz, P., and Boulanger, A., Advanced Mathematics from an Elementary Viewpoint: Chaos, Fractals, and Non-Linear Systems. *Proceedings, Computers and Mathematics Conference*, MIT, Cambridge, MA, June, 1989.

Feurzeig, W. and Lukas, G., Logo: A Programming Language for Teaching Mathematics. *Educational Technology*, 1972.

Feurzeig, W., Lukas, G., and Lukas, J. D., *The LOGO Language — Learning Mathematics Through* Programming, Entelek, Newburyport, MA, 1977.

Feurzeig, W., Lukas, G., and Stefanescu, D., Compuer-Based Interactive Flight Training. Final Report, Bolt Beranek and Newman, Cambridge, MA., November, 1980.

Feurzeig, W., Munter, P. K., Swets, J. A., and Breen, M.N.. Computer-Aided Teaching in Medical Diagnosis, *Journal of Medical Education*, Vol. 39, No. 8, 1964.

Feurzeig, W., and Papert, S., Programming-Languages as a Conceptual Framework for Teaching Mathematics. *Proceedings of the NATO Conference on Computers and Learning*, Nice, France, 1968.

Feurzeig, W., Papert, S., Bloom, M., Grant. R., and Solomon, C., Programming-Languages as a Conceptual Framework for Teaching Mathematics. Final Report on the first fifteen months of the Logo project, BBN Report No. 1989, Bolt Beranek and Newman, Cambridge, MA., November 1969.

Feurzeig, W., and Richards, J., Intelligent Tools for Algebra, *Technology and Learning, 2* (31), 1988.

Feurzeig, W., and Ritter, F., Understanding Reflective Problem Solving. In *Intelligent Tutoring Systems: Lessons Learned*. (Psotka, Massey, and Mutter, eds.) Lawrence Erlbaum, Hillsdale, N.J., 1988.

Feurzeig, W., and White, B., An Articulate Instructional System for Teaching Arithmetic Procedures. BBN Technical Report, Bolt Beranek and Newman, Cambridge, MA., January, 1984.

Frederiksen, J., and White, B., Implicit Testing Within an Intelligent Tutoring System. *Machine-Mediated Learning*, Vol. 2 No. 4, 1988.

Frederiksen, J., and White, B., An Approach to Training Based Upon Principled Task Decomposition. *Acta Psychologica*, 71 (1–3), 1989.

Gifford, C., and Adams, M. J., Reading Aide Video. `http://www.iment.com/maida/tv/songvids/mp4/computer/ReadingAide.mp4`.1996.

Goldenberg, E,P,. Learning to Think Algebraically: A Logo and Function Machines Contribution to Solving Word Problems. *Intelligent Tutoring Media*, Vol. 4, No. 3/4, Learned Information Ltd., Oxford, U. K., November 1993.

Goldenberg, E. P., and Feurzeig, W., *Exploring Language with Logo*. MIT Press, Cambridge, MA., 1987.

Grignetti, M. C., Gould, L., Hausmann, C. L., Bell, A. G., and Harris, G., Mixed-Initiative Tutorial System to Aid Users of the On-Line System (NLS). BBN Report 2969, Bolt Beranek and Newman, Cambridge, MA., 1974.

Harvey, B. *Computer Science Logo Style*. MIT Press, Cambridge, MA., 2nd Edition, Feb. 1997.

Hollan, J. D., Hutchins, E. L., and Weitzman, L., STEAMER: An Interactive Inspectable Simulation-Based Training System. *AI Magazine 5* (2), 1984.

Horwitz, P. and Eisenberg, M., MultiMap: An Interactive Tool for Mathematical Experimentation. BBN Report 7651, Bolt Beranek and Newman, Cambridge, MA, September 1991.

Horwitz, P. and Feurzeig, W., Computer-Aided Inquiry in Mathematics Education. *Journal of Computers in Mathematics and Science Teaching, 13*(3), 1994.

Horwitz, P., Neumann, E., and Schwartz, J., Teaching Science at Multiple Levels: the GenScope Program. *Communications of the Association for Computing Machines*, Vol. 38 No. 8, 1996.

Horwitz, P., and White, B., Computer Microworlds and Conceptual Change: A New Approach to Science Education. In *Improving Learning: New Perspectives* (Ramsden, Ed.) Kogan, London, 1988.

Hunter, B., Internetworking and Educational Reform: The National School Network Testbed. a BBN report, `http://www.isoc.org/HMP/PAPER/065/html/paper.html`, Bolt Beranek and Newman, Cambridge, MA, 1995.

Kalikow, D., and Swets, J. A., Experiments with Computer-Controlled Displays in Second Language Learning. *IEEE Trans Audio Electro-Acoustics.* Vol. AU-20, No. 1, March 1972.

Kurland, L. C., Granville, R., and MacLaughlin, D., Design, Development, and Implementation of an Intelligent Tutoring System for Training Radar Mechanics to Troubleshoot. In *Intelligent Instruction by Computer* (Farr and Psotka, eds.) Taylor and Francis, Washington DC, 1992.

Lewis, P,G., *Approaching Precalculus Mathematics Discretely.* MIT Press, Cambridge, MA., 1990.

Licklider, J. C R., Programmed Learning and Computer-Based Instruction. *Proceedings of the Conference on Application of Digital Computers to Automated Instruction.* (J. E. Coulson, ed.) pp. 217–239. John Wiley & Sons, Inc., New York: 1962.

Lukas, G., Berkovitz, J., and Feurzeig, W., Linear Approximation Techniques for Fitting Flight Performance Data. BBN Report 3615, Bolt Beranek and Newman, Cambridge MA, July 1977.

Lukas, G., Feurzeig, W. and Cohen, D., Research on High-Level Adaptive Training Systems, BBN Report 3193, Bolt Beranek and Newman, Cambridge MA, November 1975.

Lukas, G., and Lukas, J., *LOGO: Principles, Programming, and Projects.* Brooks/Cole Publishing Co., Monterey, CA., 1986.

Massey, L. D., Feurzeig, W., Downes-Martin, S., and Ritter, F., TRIO to INCOFT Adaptation Study. BBN Report 6194. Bolt Beranek and Newman, Cambridge MA, 1985 .

Massey, L. D., Kurland, L. C., Tenney, Y., DeBruin, J., and Quayle, K. C., Hawk Mach-III Intelligent Maintenacnce Tutor Design Developoment. BBN Report 6315, Bolt Beranek and Newman, Cambridge MA, Sept.1986.

Massey, L. D., DeBruin, J., and Roberts, B., A Training System for System Maintenance. 1988. In *Intelligent Tutoring Systems: Lessons Learned.* (Psotka, Massey, and Mutter, eds.) Lawrence Erlbaum, Hillsdale, N.J., 1988.

Morrison, D., The Co-NECT School: An Integrated K-12 Design. *Education Reform Digest.* 1997.

Myer, T. H., *Manual for Using TELCOMP Computation Service*, Bolt Beranek and Newman, Cambridge MA., October 1966.

Neumann, E., and Feurzeig, W., Object-Object Transformation Language in Science and Mathematics, Book chapter in Feurzeig and Roberts (Eds,) *Modeling and Simulation in Precollege Science and Mathematics Education.* Springer-Verlag, New York, NY, 1999.

Newman, D., Interpreting and Intelligent Tutor's Algorithmic Task: A Role for Apprenticeship as a Model for Instructional Design. *AI and Society.* Springer-Verlag, London, 1991.

Newman, D., School Networks: Delivery or Access. *Communications of the Association for Computing Machinery*, May, 1993.

Newman, D., Grignetti, M., Gross, M., and Massey, L. D., Intelligent Conduct of Fire Trainer: Intelligent Technology Applied to Simulator-Based Training in Intelligent Instruction, in *Intelligent Instruction by Computer* (Farr and Psotka, eds.), Taylor and Francis, Washington DC, 1992.

Nickerson, R. S., Characteristics of the speech of deaf persons, *The Volta Review*, pp. 342-362, 1975.

Nickerson, R. S., Kalikow, D. N., and Stevens, K. N., Computer-Aided Speech Training for the Deaf. *Journal of Speech and Hearing Disorders*, 41, 1976.

Nickerson, R. S., and Stevens, K. N., Approaches to the Study of the Relationship Between Intelligibility and Physical Properties of Speech. In J. Subtelny (Ed.), *Speech Assessment and Speech Improvement for the Hearing Impaired* (pp. 338–364). Washington, DC: A. G. Bell Association for the Deaf, 1980.

Nickerson, R. S., Stevens, K. N., Rollins, A. M., and Zue. V. W. Computers and Speech Aids. In Hochberg, Levitt, and Osberger (Eds.), *Speech of the hearing impaired: Research, training and personnel preparation* (pp. 313–324). Baltimore, MD: University Park Press, 1983.

Panagos, J., Feurzeig, W., and Ritter, F., TRIO System Documentation. BBN Report to Naval Training Systems Center, Orlando, FL., Bolt Beranek and Newman, Cambridge MA, June, 1987.

Quillian, M. R., Semantic Memory, Book chapter in *Semantic Information Processing*, M.Minsky (Ed.), MIT Press, Cambridge, MA, 1968.

Ritter, F., OREO: Orienting Electrical Circuits for Qualitative Reasoning. BBN Report 6560, Bolt Beranek and Newman, Cambridge MA, 1987.

Ritter, F., and Feurzeig, W., Teaching Real-Time Tactical Thinking. In *Intelligent Tutoring Systems: Lessons Learned.* (Psotka, Massey, and Mutter, eds.) Lawrence Erlbaum, Hillsdale, N.J., 1988.

Rubin, A., Rosebery, A. S., and Bruce, B., Elastic and Reasoning Under Uncertainty. BBN Report No. 6851, Bolt Beranek and Newman, Cambridge MA, June 1988.

Stevens, A.L, and Collins., The Goal Structure of a Socratic Tutor. *Proceedings, Association for Computing Machinery Annual Conference*, 1977.

Stevens, A. L., Roberts, B., Stead, L., Forbus, K. D., Steinberg, C., and Smith, B., STEAMER: Advanced Computer-Aided Instruction in Propulsion Engineering. BBN Report 4702, Bolt Beranek and Newman, Cambridge MA, 1981.

Stevens, A. L., Roberts, B., Stead, L., The Use of a Sophisticated Graphics Interface in Compuoter-Assisted Instruction. *IEEE Explore*, March/April 1983.

Stevens, K. N., Nickerson, R. S., Boothroyd, A., and Rollins, A., Assessment of nasality in the speech of deaf children, *Journal of Speech and Hearing Research, 19*, pp. 394-416, 1976.

Swets, J. A., Some Possible Uses of a Small Computer as a Teaching Machine, Insert in Feurzeig, Conversational Teaching Machine, *Datamation*, vol. 10 No. 6, pp. 38–42, 1964.

Swets, J. A., and Feurzeig, W., Computer-Aided Instruction. *Science*, Vol. 150, No. 3696, pp. 572–576, October, 1965.

Tenney, Y., and Kurland, L. C., The Development of Troubleshooting Expertise in Radar Mechanics. In *Intelligent Tutoring Systems: Lessons Learned.* (Psotka, Massey, and Mutter, eds.) Lawrence Erlbaum, Hillsdale, N.J., 1988.

Webster-Merriam Dictionary, 1923.

White, B., and Frederiksen, J., QUEST: Qualitative Understanding of Electrical System Troubleshooting. *Association for Computing Machinery Sigart Newletter*, July 1985.

White, B., and Frederiksen, J., Intelligent Tutoring Systems Based Upon Qualitative Model Evolutions. *Proceedings, AAAI-86, American Association for Artificial lntelligence.* 1986.

White, B., and Frederiksen, J., Causal Model Progressions as a Foundation for Intelligent Learning Environments. *Artificial Intelligence 42*(1), 1990.

White, B., and Horwitz, P., ThinkerTools: Enabling Students to Understand Physical Laws. *Proceedings, Ninth Annual Meeting of the Cognitive Science Society* (pp.336–347). Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1987.

Wight, S., Feurzeig, W., and Richards, J., Pluribus: A Visual Programming Environment for Educational Research. Proceedings, *IEEE Workshop on Language and Automation*, College Park, MD, The Computer Society, Institute for Electrical and Electronic Engineers, 1988.

Chapter 14

Speech Processing at BBN

John Makhoul

This chapter of BBN's speech processing activities covers primarily a period that began around 1971 through the early 1990s. Areas of importance—technical as well as historical—include speech recognition and understanding, speech coding, speaker recognition, and speech modification. A number of today's best-regarded techniques in speech and language processing stem from BBN's early work. Research and development in these areas continue to be a primary focus to this day. In addition, since the early 1990s, the application of speech and language technologies for commercial and government applications has gained in importance.

14.1 Before 1971

Bolt Beranek and Newman Inc. (BBN) started as an acoustics consulting company and, from its early years, the company did a good bit of work relating to amplification systems, effects of noise on people (e.g., people riding in airplanes or living near airports), and issues of intelligibility of speech communication systems (e.g., understanding what was being said over a loudspeaker system in a noisy environment or understanding person-to-person communication in a noisy military environment such as in an Army helicopter). A separate activity, begun in the 1960s, dealt with the processing of speech signals for data compression or recognition purposes in which a computer recognizes the words spoken by someone.

A brief look through the BBN Reports archive suggests that before 1971 three people helped BBN move toward the areas of speech processing described in this chapter. These are Karl Kryter, Ken Stevens, and Dan Bobrow. Each man had a connection with J. C. R. Licklider who, according to a well told story,[1,2,3] introduced digital computing to BBN and started the Information Processing Techniques Office of the U.S. Defense Department's Advanced Research Projects Agency (ARPA/IPTO).

Ken Stevens began working with BBN before Kryter or Bobrow. Trained originally in physics in Canada, Stevens arrived at MIT to get his PhD, and at MIT came under the influence of Leo Beranek who was Stevens' thesis supervisor (Licklider was also on Stevens' thesis committee). After receiving his PhD, for three or four years Stevens worked half-time at BBN and half-time as a staff member of MIT's Acoustics Lab that was co-directed by Leo Beranek and Dick Bolt. In 1956, about the time Beranek resigned from MIT to spend full time at BBN, Stevens joined the MIT faculty, but stayed on at BBN as a one-day-a-week consultant, until almost 1990. Over his years working part-time at BBN, Stevens primarily dealt with speech and hearing in various contexts (e.g., working on speech aids for deaf children on a multi-year project with Ray Nickerson). In 1969, Dick Bolt and Ken Stevens participated on a panel established by the Acoustical Society of America to report on the reliability of identifying a person, for legal purposes, by examining the spectrographic (frequency/time) patterns of his or her speech sounds. The resulting report put into question some of the exaggerated claims that were being made at the time by some practitioners about the reliability of using spectrograms for speaker identification.[4]

Karl Kryter was a close personal friend of Licklider; they had been graduate students together at the University of Rochester. Kryter had worked with Licklider in Harvard's Psycho-Acoustic Laboratory, and Kryter was best man at Licklider's wedding. Leo Beranek also knew both Licklider and Kryter from his time as director of Harvard's Electro-Acoustic Laboratory, which collaborated with the nearby Psycho-Acoustic Laboratory. After Beranek went to MIT to co-direct the Acoustics Laboratory (with Dick Bolt), he recruited Licklider to join them at MIT. Later Beranek recruited Licklider to join BBN.[5] Shortly after Licklider joined BBN in 1957, Kryter also joined BBN to work in the psycho-acoustics area.[1]

Danny Bobrow was a graduate student at MIT when Licklider recruited him in 1962 to participate in the Libraries of the Future Project,[2] and he continued to work part time at BBN after the project finished. When Bobrow finished his PhD in 1964, he had several opportunities[6] but he chose to accept Jerry Elkind's offer to restart an Artificial Intelligence (AI) group at BBN.[7]

Karl Kryter worked on many of BBN's traditional acoustics projects, particularly speech intelligibility. In 1958, he also began to study speech compression, under a contract from the U.S. Army Electronics Research and Development Laboratory. His approach was to use narrow band filters with the goal of transmitting speech at one-half or one-third of normal speech bandwidth with the intelligibility of uncompressed speech.[8] From 1958 through 1963, Kryter and colleagues (J. H. Ball, J. F. Colaruotolo, J. Melaragni, S.C. Mowry, E. Whitman, and R. Miller) studied this area and built a speech compression system based on narrow-band spectrum sampling.[9] Ball did far and away the most work on this project, and apparently the project worked reasonably well at a 4800 bps (bits per second) rate but with reduced intelligibility.

In 1962 and 1963, under contract to Rome Air Development Center, Kryter and Ken Stevens wrote several extensive reports (individually and together) evaluating approaches to speech compression.[10] After this, Kryter's work at BBN moved back to issues of intelligibility and other more traditional BBN acoustics research and development.

After he came to BBN full time and was building his AI group, Dan Bobrow got involved in the development of a limited speech recognition system (called LISPER) under contract to NASA.[11] This system was built to handle 60 or 70 words and was written in LISP, with an appropriate bank of filters and sampling to turn the speech into something the computer could work on. Working on this project with Bobrow was Dennis Klatt (like Ken Stevens, Dennis Klatt of the MIT staff consulted to BBN one day a week for many years).[12] The actual recognition was done using Warren Teitelman's ARGUS program for recognizing hand-drawn characters[13] adapted to speech patterns.

In the BBN Report archive, we find additional work by Stevens and colleagues: on speaker authentication techniques[14] and on a bank spectrum analyzer for a speech recognition system.[15]

As 1970 approached, Bobrow saw the potential for BBN to participate in an ARPA-funded speech recognition and natural language understanding project that was on the horizon. Bill Woods had already joined BBN to work in the natural language understanding area.[7,16] Bobrow sought scientists and engineers who could pursue the speech side of this opportunity to join Bill Woods and his colleagues who would work on the natural language side of the project.

Upon the recommendations of Ken Stevens and Dennis Klatt, Bobrow hired me in October 1970 specifically to work in the speech area. I had just finished my PhD at MIT where Stevens was my unofficial thesis advisor.[17] Stevens actually was Jerry Wolf's thesis supervisor at MIT; and, in 1971, after a post-doc year at the University of Edinburgh, Jerry Wolf joined the fledgling speech group at BBN.[18]

14.2 ARPA Speech Understanding Research program, 1971–1976

Speech understanding had been part of what the ARPA Information Processing Techniques Office (IPTO) meant by intelligent systems from the time Licklider founded the IPTO office in 1962. During the 1960s, several laboratories were doing work in speech recognition, including the LISPER project at BBN mentioned in the last section. The results of these projects encouraged ARPA to make a breakthrough push in speech-understanding capability, as described in the book by Norberg and O'Neill[3] (pp. 232–233). About 1970, IPTO sponsored a study regarding speech recognition and natural language understanding led by Allen Newell.[19] A National Research Council report at the time said,[20]

> ARPA established the Speech Understanding Research (SUR) program to develop a computer system that could understand continuous speech. Lawrence Roberts initiated this project in 1971 while he was director of IPTO. . . Roberts wanted a system that could handle a vocabulary of 10,000 English words spoken by anyone. His advisory board, which included Allen Newell and J. C.Ṙ. Licklider, issued a report calling for an objective of 1,000 words spoken in a quiet room by a limited number of people, using a restricted subject vocabulary (Newell *et al.*, 1971).
>
> Roberts committed $3 million per year for 5 years, with the intention of pursuing a 5-year follow-on project. . .

Speech recognition can be thought of as turning a stream of spoken audio into a stream of text words. At the time, the speech recognition accuracy for continuous speech (where the words are spoken continuously and not separated by pauses) was quite low, even for small vocabularies. Thus, the Newell-led study report suggested that speech recognition could be more successful if it could take advantage of higher-order contextual information, in the form of a grammar (with a defined syntax and semantics). Study group member Dennis Klatt is quoted on page 233 of the Norberg-O'Neill book as saying,[3]

> [We] believed that the hope for the program lay in analyzing speech within the context of specific tasks that employed strong grammatical constraints, as well as strong semantic and dialogue constraints, so that many sources of knowledge could be brought to bear to attain successful understanding of what was said or intended by the speaker.

Thus, the ARPA study recommended that ARPA fund a program in which both speech recognition and natural language understanding would be used jointly, with the objective of not only recognizing what sequence of words was spoken but also understanding the meaning of what was said. Thus was born the ARPA Speech Understanding Research (SUR) program.

In 1971, Bill Woods and I wrote BBN's proposal for funding under ARPA's SUR program, and the BBN proposal was funded along with proposals by Carnegie Mellon University (CMU), MIT's Lincoln Laboratory, Stanford Research Institute (now SRI International), and System Development Corporation (SDC).[21] In addition to the five major sites who were slated to build complete speech understanding systems, ARPA funded a number of other sites to work on various specific research topics. Bill Woods was principal investigator for the BBN effort and led the natural language understanding part of the project, which was written in LISP. I led the speech recognition part of the project, which was written mostly in Fortran and BCPL, with some library functions written in PDP-10 assembly language.

Key participants with me on the speech recognition task were Rich Schwartz and Jerry Wolf. Rich Schwartz came to BBN after getting his B.S. degree from MIT. His undergraduate thesis had been in the speech processing area and was supervised by Dennis Klatt. In addition to his research efforts, Wolf took care of the systems aspects of the project (at MIT, Wolf had been significantly responsible for creating the PDP-9-based capability used by Ken Stevens' group). Schwartz concentrated on recognition algorithms for the project.[22]

For BBN's ARPA SUR project of the 1970s,[23] the system comprised four stages: feature extraction, segmentation, labeling (or recognition), and word matching. In the feature extraction stage, the speech input was first sampled and digitized at 20,000 times a second. Then, for every frame of 10 ms, a number of parameters were extracted from the digitized signal. Using linear prediction analysis (see sidebar on page 357), the speech formants (resonances of the vocal tract) were extracted by locating the peaks in the spectrum. Other parameters included the number of zero crossings of the speech signal in the frame, and the energy in various spectral bands.

After feature extraction came segmentation, whereby the speech was segmented into possible phonetic segments using a set of rules, derived by looking at features extracted from a representative sample of speech data. The ranges of parameter values in the rules, as well as any thresholds, were derived by collecting statistics from hand-segmented data. The different segmentation possibilities formed a *segment lattice*.

After segmentation came labeling, which was done in two parts. First, a set of rules was used to assign each phonetic segment into one of a few broad phonetic classes — vowels, nasals, stops (p,t,k,b,d,g), fricatives, etc. Then, the specific phoneme was chosen using a set of statistical classifiers, based on features measured during the phoneme, and derived from a substantial set of hand marked speech. For each segment in the lattice, then, a single phoneme was assigned.

After phonetic labeling came word matching, where the recognized phonemes in the lattice were matched against the words in a phonetic dictionary (of the allowable words in the application). In order to make the matches more flexible, and to allow for possible phonetic recognition errors, a *confusion matrix* was used, which gave for each phoneme the probability that that phoneme might be confused with each of the other phonemes in the system. (Such a matrix was estimated from error statistics collected for the system.) Thus, instead of requiring exact phoneme matches, probabilities were computed for all possible word matches and the sequences with the highest probabilities were then passed on to the natural language part of the system which tried to make sense of the words using syntactic and semantic rules, and ultimately a single sentence of words was chosen as the output of the recognizer. Jack Klovstad at BBN wrote much of the word matching and search parts of the system.[24]

ARPA's SUR program was supposed to narrow down the list of participants from five to three participating groups after three years. However, only the Lincoln Laboratory effort was dropped; the SDC and SRI efforts were combined into a single project, leaving four groups working on three projects.

In 1976, BBN, CMU, and SDC-SRI demonstrated systems (CMU demonstrated two systems). The BBN system, called HWIM (Hear What I Mean), was able to recognize a sentence from the travel task[25] environment in two hours on a PDP-10 (that significant

Digital Sampling and Speech Analysis

Speech processing systems, whether the goal is recognition or something else, typically start by converting an analog signal representing the speech (i.e., the signal that comes out of a microphone into which the speaker speaks) into digital samples. A typical sampling rate might be 8 kHz (8,000 samples per second) or 16 kHz, depending on whether the original signal was narrow bandwidth (e.g., telephone bandwidth is only up to about 3,500 Hz) or a signal with a wider bandwidth, such as that recorded by a regular microphone. Once the analog input signal is digitized and stored in a computer, much can be done with it.

Among the techniques BBN has focused on over the years to process the digitized speech are linear prediction modeling and cepstrum analysis.

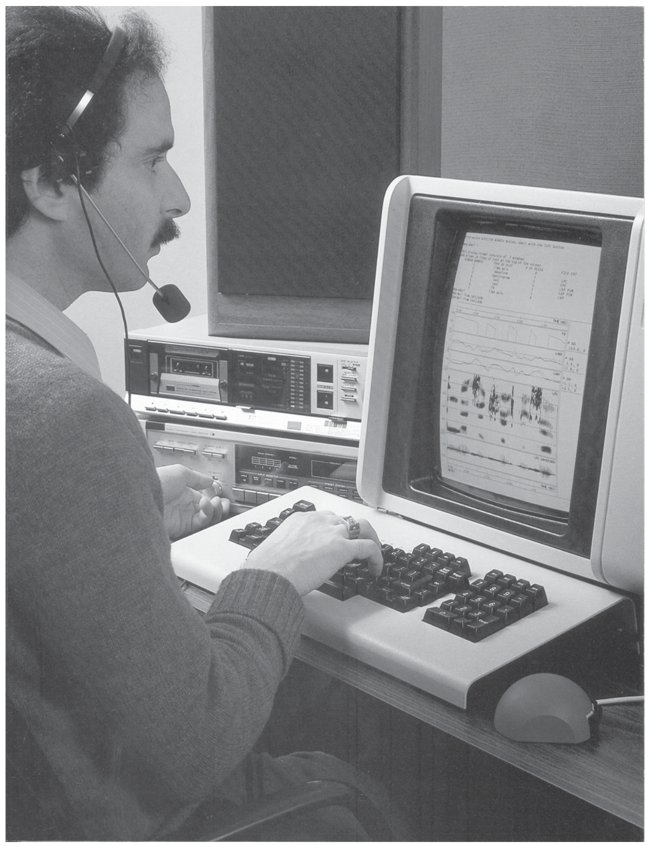


Figure 14.1. Photograph of Rich Schwartz looking at a screen of processed speech, taken for the 1984 BBN Annual Report.

Linear Prediction. In linear prediction, the digital speech signal is modeled as a weighted linear summation of the preceding dozen or so samples. The values of the weights are estimated, about every 10 ms, by minimizing the mean-squared error between the predicted value and the original signal over a window of about 20 ms. In the spectral (frequency) domain, linear prediction is equivalent to assuming that the speech is the output of an all-pole digital filter that consists only of resonances (no anti-resonances). Such a model is approximately valid for many speech sounds, especially vowels. The frequency locations of many of the peaks of the corresponding spectrum are good approximations to the locations of the natural resonances of the human vocal tract (i.e., the formants).

Linear prediction for speech processing was developed chiefly by B. Atal at Bell Labs in the late 1960s and simultaneously by F. Itakura at NTT in Japan. BBN and many other groups have used the technique in their speech processing systems. My involvement in linear prediction work started in late 1971 when I wrote a conference paper on the spectral properties of linear prediction. Upon a request by Danny Bobrow to explain the paper I had written to a less specialized audience (Danny's expertise was Artificial Intelligence), I embarked on a research effort to understand better and explain linear prediction concepts. The result, after six months of work, was a 237-page BBN report on the subject.[26] That report then formed the basis for my tutorial review paper on linear prediction.[27] It was through the writing of the BBN report and the tutorial review paper that I appreciated the importance of writing as a forcing function to improve one's understanding of a topic.

BBN's HWIM system for the ARPA SUR program (discussed in section 2) used linear prediction to model the speech for the purpose of locating the speech formants. Some of BBN's speech coding projects (described in section 3) also used linear prediction.

Cepstrum. The cepstrum[28] is simply the inverse Fourier transform of the logarithm of the spectrum. The first dozen coefficients or so have been shown to be effective parameters for speech recognition purposes.[29] BBN's use of the cepstrum first computes the log of the power spectrum (the amount of energy at each frequency) on a nonlinear frequency scale, which gives more weight to the lower frequencies to reflect the fact that perception is more discriminating at lower frequencies, then does an inverse Fourier transform. Using about a dozen of the resulting coefficients, the overall shape of the spectrum is modeled without modeling unwanted details (like pitch).

---

parts of the system were written in LISP surely didn't help system speed). The CMU systems performed better in terms of accuracy (and met ARPA's goals better); however, the BBN system undertook a more difficult task (the branching factor[30] was six times that of CMU's tasks), so it was "unclear whether there [were] large differences in ability"[31] among the CMU and BBN systems. In any case, neither system performed in real time, which George Heilmeier (overall ARPA director from 1975 to 1977) insisted was necessary for a speech understanding system to be relevant; thus, there was no follow-on program.

Computing speed. In the SUR program, computing speed was never viewed as an important issue. Because speech understanding was such a new area of research, the emphasis was largely on developing the technology and making things work, with the belief that speed could always be achieved later on through a combination of software and hardware development, as needed. Furthermore, the SUR program was seen largely as an AI problem and, at the time, it was believed that you needed to use computer languages like LISP to deal with such problems, especially for the natural language aspects of the problem. It was not obvious then, but it has now become very clear that effective research can only be accomplished if the turnaround times of running meaningful experiments are on the order of hours or maybe days, not weeks, in order to make effective use of staff time. At the time of the SUR program, statistically significant experiments were simply not feasible.

Much was learned and discovered in the SUR program. However, it is this author's opinion that, even if the systems were to have run in real-time, the technology that was developed could have supported only the simplest of applications. The world had to wait for a shift in the speech modeling paradigm (see section 14.5), as well as considerable advances in computing to support the computational intensive nature of the new paradigm, before significant advances in speech recognition were to be made.

14.3 Speech coding and compression, 1972–1991

In 1972 — a year into the ARPA SUR program — with the ARPANET[32] newly available, ARPA started seedling efforts at BBN and elsewhere that would eventually turn into the ARPA Network Speech Compression (NSC) program, aimed at developing the technology to transmit speech digitally over the network. In addition to BBN, participants in the NSC program included Lincoln Laboratory, SRI, USC-ISI, Culler-Harrison (CHI), Speech Communication Research Laboratory (later Signal Technology), and the University of Utah. The groups were instructed to work collaboratively, with each group concentrating its work in certain areas. For instance, Danny Cohen at ISI and Cliff Weinstein at Lincoln Laboratory and their groups mostly concerned themselves with network transmission algorithms, while we at BBN mostly concerned ourselves with compression algorithms.

Others participated in the BBN project, but the key BBN person was Vishu Viswanathan. Vishu came to BBN in 1972 with a PhD in control theory from Yale. He was hired by Shelly Baron[33] to work on the ARPA speech compression effort and was the technical lead on many of BBN's speech compression and coding efforts for the following fourteen years.

At the time, the baseline for digital telephony was 64 kbps (kilobits per second), which is obtained by sampling the speech signal at 8 kHz, with 8 bits per sample. The digitization to 8 bits is performed in a quasi-logarithmic manner, placing more bits at smaller values of the signal. The 64 kbps digital speech maintained the speech quality and intelligibility of the original analog telephone signal, which was termed as toll quality speech.

However, the ARPANET in the 1970s used 50 kbps phone links, which were also used for other network traffic. Thus, the baseline 64 kbps of speech data had to be compressed into fewer bits for transmission over the network. Part of the ARPA NSC program was, therefore, devoted to compressing the digital speech rate as much as possible, while trying to maintain its quality and intelligibility. A range of data rates was tried, with preference for average rates around 2400 bps, since at such rates the speech was still of reasonable quality and intelligibility, while much lower data rates reduced the speech quality and intelligibility significantly. The actual data compression was done at a variable rate, where data was transmitted only when there were significant changes in parameter values.[34]

Another part of the NSC program was devoted to the mechanics and issues associated with transmitting a real-time signal over a packet-switched network. The ARPANET (and networks since then) was based on packet switching, so speech data had to be packetized into 1,000 bit packets which might follow different paths with different delays through the network and arrive at the destination out of order or with some packets missing entirely. Thus, real-time speech communication over the ARPANET required more at the destination site than simply decoding the encoded speech data and using it to resynthesize the speech. Buffering had to be provided to try to reorder the sequence of packets that had been sent from the source, but not so much buffering that the additional delay was distracting to the human user. Decisions had to be made relating to the trade-off between missing data and late data and which sounds worse, i.e., whether to discard a packet that didn't arrive within a certain time after the preceding packet.[35,36] And so on. The first real-time demonstration of two-way packet speech communication using compressed speech took place between CHI and Lincoln Laboratory in December 1974.[37,38]

To provide 2400 bps speech over the ARPANET, the BBN system used linear predictive coding (LPC) to compute and transmit the short-term spectral parameters every

20 ms. As part of the analysis, voicing was also detected and a pitch period calculated and transmitted if the sound was voiced (like the vowels). Otherwise, the pitch period was not sent, and white noise of the appropriate level was used at the decoder end to resynthesize the speech.

The ARPA NSC Program lasted till 1982; the final meeting of the program took place in June of that year at the MIT Lincoln Laboratory. During that meeting, a live demonstration of digital voice transmission over a combination of packet-switched networks (including Packet Radio Net) took place between Lincoln, SRI, and USC-ISI.[39] At that point, the problem was largely understood and solved well enough to be useful, and ARPA's NSC program ended.[40,41]

In 1976, as it became clear that the ARPA SUR program was coming to an end, BBN sought and received funding that expanded the speech compression activity significantly over the following decade. Some clients wanted better quality than was possible with a 2400 bps system, other clients wanted lower bit-rate systems, and yet others wanted reduced degradation in the face of channel errors. Thus, a series of systems was developed at various data rates, including those that protected the data against channel errors. In one of the projects, the variable rate speech compression developed for the packet network application was adapted for fixed-rate 2400 bps transmission over a noisy channel through the use of a buffer-controlled, variable-to-fixed rate conversion mechanism.[42]

Systems that transmitted speech at 9.6 kbps[43] or 16 kbps[44] were developed, with the speech at the higher bit rate achieving toll quality. These systems used residual-excited coding — computing the linear prediction filter every 20–30 ms, inverse filtering to produce the residual, quantizing the residual using about 1–2 bits per sample, and transmitting the quantized residual as well as the LPC coefficients. Error protection was used to protect the LPC coefficients against channel transmission errors.

A number of speech compression projects at 2.4, 9.6, and 16 kbps[45] were funded by the Defense Communications Agency (DCA) during the period 1979–1986 and were led by Vishu Viswanathan. One unique aspect of the activity at the higher data rates was that BBN designed, built, and delivered board-level, real-time, multi-channel versions of these systems to several customers, including DCA.[46,47] Mike Krasner, who later became manager of the Speech Department at BBN, played a key role in managing the development of these real-time systems.

A different, and parallel, activity was aimed at coding speech at very low data rates in the ranges of 100–200 bps and 300–600 bps. At the higher of these data rates, essentially the same analysis as for the 2400 bps system was performed, but various methods that reduced the data rate were used, including variable frame rate and vector quantization of the LPC coefficients,[48] with each frame of 12 coefficients quantized as a single 8-bit number.

At 100–275 bps, several consecutive frames, comprising a segment of speech, were vector quantized together using about 13 bits per segment.[49] Work on the resulting *segment vocoder*,[50] which operated around 275 bps, was performed initially by Salim Roucos and later continued by Patrick Peterson and Philippe Jeanrenaud.[51]

At lower rates of about 100 bps, a different method was used to quantize speech segments: phonetic recognition.[52] Essentially, a speech recognition system was used to recognize each speech segment and the identity of the phoneme was transmitted. For the 100–200 bps coders, the pitch and energy were heavily quantized and transmitted once for each segment of speech. Needless to say, the lower the transmission rate, the lower was the speech intelligibility. For the 100 bps coder, a phonetic recognition accuracy of over 85% was needed to maintain reasonable intelligibility, but the state of the art of speech recognition at the time was such that it was not possible to achieve such high phonetic accuracy.

In the speech coding area, a number of additional technical contributions are still actively referenced by other researchers. These include: optimal quantization of LPC coefficients,[53] lattice methods for linear prediction,[54] the objective speech quality evaluation of LPC coders,[55] and a mixed-source model for speech compression and synthesis.[56]

By 1991, all speech compression work at BBN had stopped for lack of funding. Speech compression technology had matured sufficiently so that future research and development, leading to commercial products, was carried on primarily by industry.

14.4 Scrambling for work

After the ARPA SUR program stopped in 1976, the BBN speech researchers needed to seek work in addition to expanding the speech coding work (described in section 3).

Speech Modification

Several efforts were undertaken to modify the speech in various ways and for different applications. The model used in speech coding effectively decomposes the speech signal into three independent components: excitation (which includes the pitch of the voice), spectral shape (which determines which sound is spoken), and time. By modifying any or all of these three components, one can generate interesting effects upon resynthesis.

One amusing demonstration of voice modification was made by Lynn Cosell[57] and me in 1975. We took a recording of then division director Bert Sutherland (later of Xerox PARC) saying “System reliability has become an increasingly important issue” and modified his voice in the following ways. First, by changing the time dimension, the sentence was played out slower or faster in a natural way (without the funny sound effects you get by playing a tape slower or faster). Then, by doubling the pitch and stretching the spectrum by 15%, we were able to make the voice sound like a female (the female vocal tract is shorter than the male’s by about 15%). Finally, by changing just the pitch, we were able to have Bert “sing” a tune that I wrote for the occasion.

Helium speech. But then, more serious applications of voice modification emerged. Cosell and I built a system that made helium speech more intelligible.[58] To prevent bends, divers breathe air to which helium has been added, giving the divers’ speech a Donald Duck quality. This quality results from the fact that sound travels faster in Helium than in air, thereby “stretching” the speech spectrum in frequency by about a factor of 2.5. However, the voice pitch does not change because it does not depend on the speed of sound. So, by decomposing Helium speech into an excitation and a short-term spectrum, one can keep the excitation the same but compress the spectrum by the same factor of 2.5 and resynthesize. The result is much improved naturalness and intelligibilty.

Voice identity modification. Another client wanted a voice modification system, i.e., a system for making one person sound like another specific person. For this application, we make the distribution of the pitch values for the two speakers to be the same, and we also make the long-term spectrum of the two speakers to be the same. With these two changes, a large fraction of the speaker-specific characteristics are captured and the effect of the modification can be quite believable. This work was led first by Jerry Wolf and then by Vishu Viswanathan.

High-quality modification. The method for speech modification was then improved to make the speech sound more natural by including more details in the speech excitation. This was done using the speech *residual*, which is computed by flattening the short-term spectrum of the speech signal through filtering it using the inverse of the all-pole linear prediction filter. The resulting method, which was developed by Salim Roucos,[59] was called the synchronized-overlap-add (SOLA) method and has become a widely used method for the time-scale modification of speech.

Speech enhancement. In 1978, our group began work on a contract to do speech enhancement in noise. The client wanted to minimize the existing noise and enhance intelligibility. Michael Beyrouti and Rich Schwartz created an improved spectral subtraction method that reduced significantly the "musical noise" that was typical of other speech enhancement methods.[60] Even though this was a relatively short effort, this BBN work continues to be referenced as one of the successful attempts at solving the musical noise problem in speech enhancement.

Under the sponsorship of the Air Force Rome Air Development Center (RADC), we worked on the enhancement of speech in high levels of noise, especially in a fighter aircraft environment. In a series of projects from 1982 to 1988, led by Vishu Viswanathan, we experimented with the use of multi-sensor systems. One particularly successful configuration used two sensors: a noise-cancelling microphone plus a throat-mounted accelerometer that measured skin vibrations. The system was shown to have higher speech intelligibility and quality in those noisy environments than a single sensor input.[61] The two-sensor system also resulted in improved accuracy for automatic speech recognition. An exploratory development model was fabricated and delivered to RADC for their evaluation in real fighter aircraft cockpits.[62] Ken Stevens of MIT served as a consultant to these projects.

Speaker Recognition

Speaker recognition work can be partitioned into *identification* and *verification*. In identification, the system attempts to recognize the identity of a person from among a set of known individuals, based on the person's voice. In verification, the system merely decides whether a claimed identity is correct or not, again based on a sample of the person's voice.

In 1981, Rich Schwartz approached me with an idea for a new method to perform speaker identification. The result was an internally funded R&D project, where Schwartz set out to demonstrate that using sound statistical modeling principles was superior to the then current method of comparing the average spectrum of the test sample to the average spectra of the known speakers. The success of that effort[63] led to a series of funded projects that have continued, off and on, until the present, and the basic method developed at BBN became the dominant method for speaker identification. The projects were first led by Jerry Wolf, then Mike Krasner, and later Herb Gish.[64]

In order to gain a better appreciation of how speaker identification technology might be used, consider the hypothetical example where you have several people participating in a voice conference call, each person calls in to the phone company operator who is setting up the call, gives his or her name, and is connected into the call. If you further imagine that an automatic conference call transcription program is running, that program would put an identifying name with each speaker's words in the transcription of the conference call. The historical way of recognizing speakers was to store the average spectrum of each possible speaker into a database; then to compare the average spectrum of the speaker under question with the average spectrum of each speaker in the database, selecting the best match. Rich Schwartz improved on this process by comparing the probability distribution of the spectra generated by the speaker in question with the distributions for the speakers in the database, measuring the likelihoods, and picking the most likely match.

The work on speaker verification at BBN also started with an internally funded R&D effort in 1984, again led by Rich Schwartz, in anticipation of a request for proposals from the government. For verification (versus identification) you have the cooperation of the person being verified. For instance, the person presents a badge with an id number to a badge reader, and the system then verifies that the correct person has the badge and is speaking (based on prior speech samples the person has provided to the system) before allowing admittance to a secure area. The comparison of the newly presented speaker data to the data for that speaker stored in the system can be either text dependent or text independent. A common approach is for the speaker to have pre-recorded the speech of the digits from zero to nine. Then, when attempting admittance, the speaker says a random set of digits presented to him by the system or may be requested to give his or her numeric password, and what the speaker says can be checked both for being the correct speaker and for speaking the digits in the correct order.

BBN's approach used two innovations that, together, achieved a significant improvement in the then state of the art. First, the models of the speakers' voices stored in the database were hidden Markov models, or HMMs (see sidebar on page 364), which were quite new at the time and had previously not been used in speaker verification. The advantage of using HMMs here was that they provided a time-based probabilistic model of what was spoken. However, using HMMs was not enough. It is possible that a speaker-to-be-identified can be a good match to someone in the database but not be a person in the database. A simple minimum threshold of comparison is not sufficient since that could be data dependent. Rich Schwartz suggested a method for dealing with this problem. In addition to storing models for everyone who was registered in the database, a probabilistic model was stored that represented everyone else who was not in the database (this model was estimated from a large number of people not in the database). The odds of correct identification can be improved by testing that it is likely the speaker-to-be-identified is both a good match for one of the individuals in the database *and* not a good match for everyone else who is not in the database. The addition of the so-called "alternate model" did away with the problem of data dependence and greatly improved the accuracy of the system (the error rate was reduced by about an order of magnitude).

Following the success of the first laboratory-based project, a second internally funded R&D project was initiated to demonstrate the ideas in a real-world implementation of the system. Such a system was built and connected to the entrance to BBN's 10 Moulton Street building from its parking garage. Speakers entering the building had to punch in their telephone extension (as a form of identifying themselves) and speak one of a set of phrases. If the verification was successful, the door was unlocked and the person entered the building. The system was in place for about six months and was used regularly by about 40 volunteers.[74]

A proposal for external funding was written in 1986; but the proposal was not funded, we were told, in part because using HMMs was too new a technology, so the agency went with a more traditional approach. However, the big benefit of BBN's work on this project was sharpening the intuition of the BBN researchers regarding use of a probabilistic *alternate model* which proved valuable in later work. In particular, the idea was used in speaker identification, which allowed the system to decide whether the speaker was one of the known speakers or not. Here, in addition to estimating a model for each of the known speakers, there was also an alternate model which, effectively,

Hidden Markov Models

Hidden Markov models (HMMs) were developed starting in the late 1960s by Baum and colleagues[65] at the Institute for Defense Analyses (IDA) in Princeton, NJ. While traditional pattern recognition methods employ *static* probabilistic models defined over some feature space of interest (like spectra), HMMs are *dynamic* in that the models are a function of an independent variable (such as time). Through their ability to model variability in feature space and in time simultaneously, in the modeling of speech, HMMs are able to model phonetic variability as well as speaker variability.

One of the most important properties of HMMs is that the parameters of the models can be estimated automatically from training data, without the need for explicit alignment between the speech data and the words. For a given corpus of training speech data, one merely needs to provide the sequence of words that were spoken and a phonetic dictionary that specifies how the words are pronounced; the actual training of the models is largely automatic. Another important property of HMMs is that there is no separate segmentation of the speech into phonemes and words; the segmentation happens implicitly as part of the recognition process. This is in sharp contrast with rule-based methods where there is a separate segmentation stage, followed by recognition.

In the 1970s, IDA tried HMMs on a variety of applications including speech. Then, they decided to publicize their HMM technology to get more people in the speech community to use it and develop it further. So, they held a workshop in 1980 in Princeton to which about 40 people were invited. Vishu Viswanathan attended for BBN and brought home with him a preprint of a little book ("the blue book") IDA had prepared which provided the theory of HMMs and some example applications. Later, IDA decided not to publish the blue book, but photocopies of the book were already being made and distributed.

Prior to 1980, Jim Baker had used HMMs in his Dragon speech recognition system, which formed his PhD thesis at CMU.[66,67] At about the same time, Fred Jelinek and his colleagues at IBM had formulated the speech recognition problem from a statistical, information theoretic point of view[68], where speech is viewed as the output of an encoder (the human) and speech recognition, therefore, as a decoder whose objective was to decode the encoded sequence of words.[69] Later on, HMMs and the IBM new formulation were joined into a powerful new paradigm for speech recognition that still forms the backbone of all state-of-the-art speech recognition systems.[70]

After the IDA workshop in 1980, the move to using HMMs in the speech community occurred relatively slowly. AT&T started using HMMs[71] and Larry Rabiner later on wrote a definitive tutorial review about HMMs.[72] BBN's involvement in HMMs started in 1983, as noted at the end of section 4. The widespread use of HMMs for speech recognition did not happen until after 1986, as noted in section 5.[73]

modeled all other speakers. To develop this alternate model, one had to collect data from tens of speakers, which was adequate for many applications.

Transition Back into Speech Recognition

The transition back into speech recognition and into the modern era of performing recognition using HMMs started in 1983 in an ARPA-sponsored project that was the precursor to the Strategic Computing Program (see next section). In this project, BBN introduced the concept of context modeling, in which the model of a phoneme was made to depend on the neighboring phonemes.[75] The problem chosen was that of the recognition of the E-set (i.e., the letters B, C, D, E, P, T, V, Z), which was thought to be a difficult recognition problem at the time.

The first project to take advantage of this humble return to the world of speech recognition was one sponsored by the Sensory Aids Foundation to develop a speech communication aid for the hearing impaired called VIDVOX.[76] The idea was to develop a device that would include the use of phonetic recognition and certain prosodic cues, which were to be displayed to the hard of hearing person to help him understand what is being said. The project, which was led by Mike Krasner and Rich Schwartz, also included Bill Huggins, Owen Kimball and Yen-Lu Chow. The project demonstrated that a higher phonetic accuracy than was possible at the time would be needed for the technology to be of utility as an aid for the hearing impaired. The project also served as a springboard for further contributions to the state of the art in speech recognition as part of a new ARPA program, as described below.

14.5 Speech recognition, since 1984

A New ARPA Speech Recognition Program

After the cancellation of the ARPA speech understanding program in 1976, BBN kept its finger in the speech recognition waters a bit, for instance, the 100 bps coding work sketched in section 3 used a form of recognition. In 1984, ARPA again began to sponsor speech recognition work as part of the Strategic Computing program.[77] This time, however, ARPA planned to let only one big contract, with small contracts going to several other groups to support the main effort. BBN bid on the big contract proposing a system based on HMMs. CMU bid on the big system using a rule-based approach,[78] as had been used by most contractors in the 1970s SUR program. CMU chose a rule-based approach based partially on impressive, then relatively recent, spectrogram-reading experiments that Victor Zue from MIT, in collaboration with Ron Cole from CMU, performed at CMU, whereby Zue was able to determine the phonetic sequence of an utterance with relatively high accuracy simply by looking at its spectrogram. CMU's approach was to codify in software the rules that Zue used in reading spectrograms and perform recognition in that manner. CMU won the big contract and BBN was given one of the small contracts to work on statistical modeling using HMMs.

At the time, there was a strong bias against statistical methods in ARPA/IPTO circles, claiming that statistical methods could not possibly capture phonetic information and that such "knowledge" had to be captured in the form of acoustic-phonetic rules. At BBN, we had seen the value of mathematically sound models. Thus, we were convinced that a system that used statistical principles based on HMMs would not only capture phonetic information very well, but that the basic HMM paradigm was fundamentally more rigorous and sound than rule-based methods, and that it would lead to far superior results. Rule-based methods, which depended on sequencing of decisions, each of which raised the possibility of irrecoverable errors, were no match to a method that made its decisions by incorporating all sources of knowledge simultaneously. Furthermore, HMMs were automatically trainable to optimize performance, and the method could be extended to other languages easily, without rewriting new rules for each language. Speaking somewhat more philosophically, Rich Schwartz and I described these ideas as "ignorance modeling":[79] since little is known about how things actually work in human beings (e.g., how humans turn sounds into words and words into sentences), that ignorance is best exploited by modeling it mathematically.

While the contract BBN received under the Strategic Computing program was not big, it was sufficient to build a complete speech recognition system using HMMs. (Curiously, ARPA required that the software be written in LISP for Symbolics machines, again demonstrating the bias of the times.) We called the BBN system "Byblos,"[80] after the ancient Phoenician city[81] where the first phonetic writing was found, precisely to counter a prevailing view in the ARPA technical community at the time that statistical methods would not be good at modeling phonemes. The key people involved in the development of the first Byblos system were Yen-Lu Chow, Owen Kimball, and Francis Kubala, with Rich Schwartz as technical lead.

The system BBN built had a feature extraction stage where the cepstrum (see sidebar on page 357) was computed every 10 ms, and the first dozen coefficients were used in the modeling and recognition. The HMM was then used to model the variability of the cepstral feature vector as a function of time, for each phonetic context. It was not sufficient to have a single model for each phoneme, because the acoustic manifestations of phonemes changed dramatically depending on the neighboring phonemes. BBN was the first to demonstrate the effectiveness of using phonetic context to improve recognition accuracy.[82]

In addition to the *acoustic model*, which modeled the speech sounds, there was also a *language model*, which represented *a priori* constraints on the words used by the system (its vocabulary) and the frequency with which different word sequences are used in the language. Associated with each word was its phonetic pronunciation, which was written by hand. As for the frequency of word sequences, it was adequate at the time to estimate the frequencies of three-word sequences (trigrams).

The power of the new HMM paradigm, relative to rule-based methods, was demonstrated for the first time in February 1986, when the first ARPA competitive evaluation took place with CMU and BBN being the only participants. Although it was written in LISP and ran very slowly, BBN's HMM-based system performed significantly better, in terms of accuracy, than CMU's rule-based system. In July of the same year, during a government project review, we gave a live demonstration of the Byblos system, which showed graphically the top scoring word hypotheses at each point in time (see Figure 14.2). The demonstration — in addition to providing entertainment to fill the two minutes it took to perform the recognition of the utterance — was key to convincing the government visitors[83] that, indeed, the HMM approach was dealing effectively with fine phonetic distinctions.[84]

It is worth pointing out that the requirement by ARPA to have its funds used to purchase LISP machines was a way for ARPA to encourage the development of the computing infrastructure that would support AI research, and LISP was the primary language used for AI research, and speech recognition and understanding were viewed as part of AI research. At the same time, ARPA was funding the development of another type of computing infrastructure — that of parallel processing computers, and BBN was engaged in developing such computers.

So, the initial effort taken by BBN to overcome the speed problem of the HMM approach was to use the BBN Butterfly parallel processing computer. Thus, at the Fall 1987 meeting of the ARPA program at BBN, BBN demonstrated the Byblos speech recognition system running on a 97-processor Butterfly computer,[85] which performed the recognition with a 1000-word vocabulary in close to real time. However, the major contributions in developing search algorithms for real-time recognition came afterwards in a series of innovations by Rich Schwartz and colleagues which began with the N-best search algorithm,[86] followed in 1990 by a real-time system implemented on a single-processor SUN,[87] and culminated in January 1993 in the demonstration, during the ARPA project meeting at MIT, of the world's first 20,000-word continuous speech recognition system running in real-time on a single-processor, off-the-shelf, HP workstation. Algorithmic speedups of two orders of magnitude, along with significant increases in computer speeds, made this feat possible. The latter work was performed jointly with Long Nguyen and was patented and published at a later date.[88,89]

It took two simultaneous developments to make real-time speech recognition on

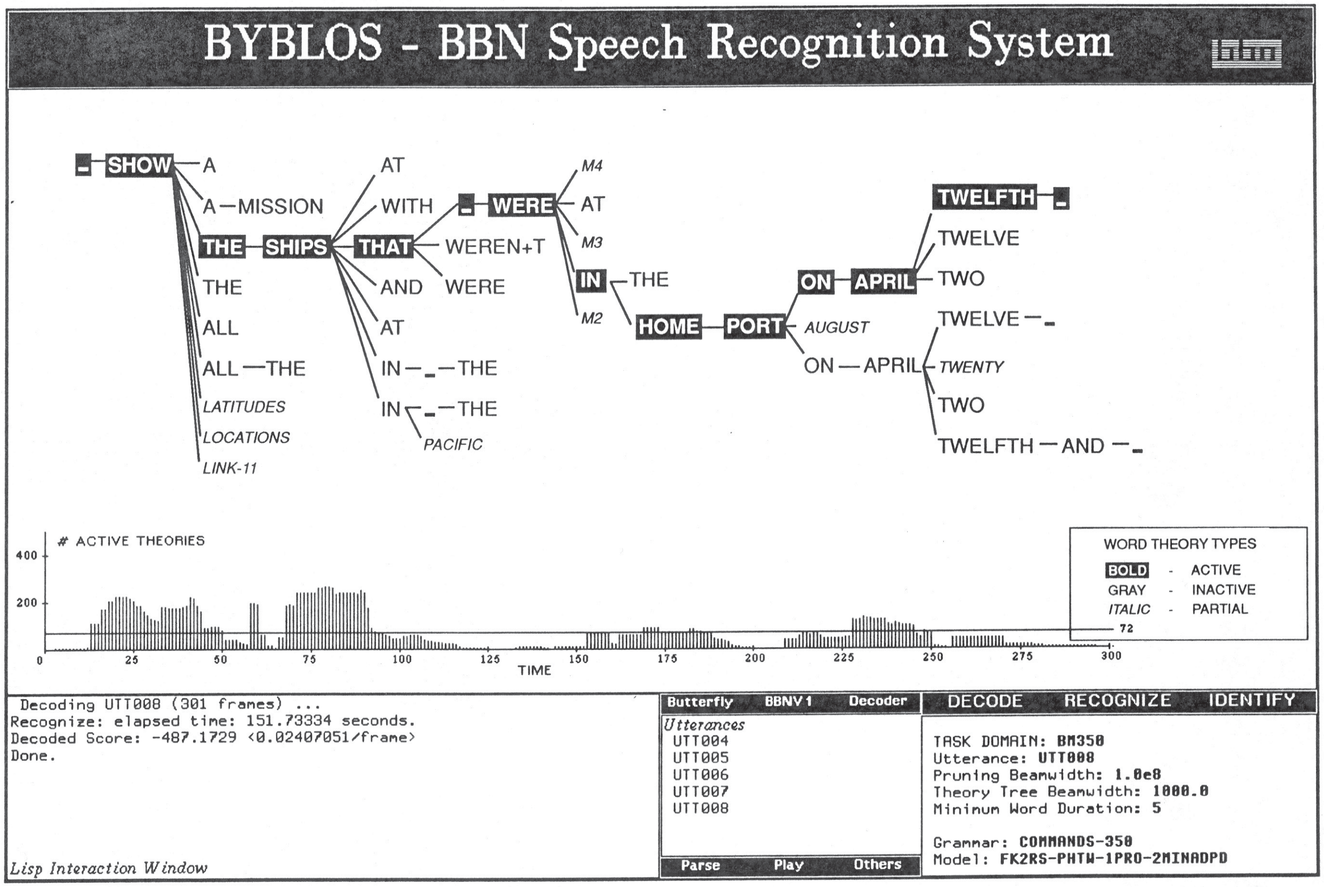


Figure 14.2. Display of the final output of the Byblos recognizer as shown in the July 1986 demonstration. The display shows that the system is making distinctions between phonetically close words, e.g., A vs. THE; AT vs. THAT; WERE vs. WEREN'T; and TWELVE vs. TWELFTH.

an off-the-shelf computer a reality: the progression of increased computer speed and memory (following Moore's Law) and the algorithmic developments that performed the recognition search efficiently without loss in accuracy. These two developments have fueled advances in speech recognition technology by allowing researchers to use more sophisticated and computationally demanding models.

CMU's primary project under their ARPA contract was using a rule-based approach, but CMU had one student (Kai-Fu Lee) working on a side project using the HMM approach (Rich Schwartz was sharing ideas with Lee). By 1988, CMU had switched their whole project over to using HMM methods.[90]

An important aspect of this ARPA program (versus the 1970s SUR program) was up-front consideration of competitive evaluations and the decision that the groups would use common training data and common test sets. That way, it was clear which system worked better. In fact, groups not part of the ARPA project could create their systems and test them using the common data and, thus, bring their systems to the program meetings, and participate in the competitive evaluations. This objective approach to evaluation made the relative value of the HMM approach clear and helped spread it to the world.

Wordspotting

In parallel with the ARPA project, and starting in 1987, BBN obtained a contract to work on wordspotting — computer detection of a small list of words in a long stream of speech. You can imagine applications where it would be useful to have a computer that could call a human's attention to a stream of speech when words in certain topic areas were spotted. The work on this project was done principally by Salim Roucos (now spelled Roukos) and Robin Rohlicek, but later work was led by Herb Gish.

The dominant approach historically had been to have models for the few words of interest (keywords) and then to do pattern matching of the models of the keywords against the speech stream. This approach caught only about 40 percent of the instances of the words in the speech stream, and that level of performance hadn't improved in over 20 years.

The problem here was the same as for some of the speaker recognition problems[91] — the thresholds used in the pattern matching were sensitive to the data. Again, the solution was to provide an alternate model — a model for all words except the keywords. Many alternate models were developed.[92] But, once it was recognized that the best model for the rest of the words is models for those words themselves, the solution was then to recognize all the words (or as many as possible) and then pick out the few keywords of interest. So, the wordspotting problem turned into a large-vocabulary speech recognition problem, which was the same goal as the ARPA program. Thus, there was much synergy between the two projects.

The wordspotting work at BBN continues to this day under the leadership of Owen Kimball.

Back to Speech Understanding

After a hiatus of 12 years, and buoyed by the success of the new HMM speech recognition paradigm, ARPA, in 1988, started a Spoken Language Systems (SLS) program to develop systems that can understand spoken dialogues. This ARPA activity, in one form or other, has continued to the present.

At the time, the speech activity at BBN was in one department (Speech Signal Processing Department) and the natural language activity was in the AI department. Allen Sears, program manager of the ARPA SLS program, suggested that BBN might want to combine the two activities into one department to facilitate the SLS work. Soon thereafter, BBN combined the two activities under a single department (Speech and Language Processing Department) with me as department manager and Madeleine Bates from the natural language activity as assistant manager. The joining of the two groups into one department has served as an important catalyst in bringing the two groups closer together, not only at a personal level, but, more importantly, at a technical level. The shift to a statistical modeling paradigm, which started with the work on speech and speaker recognition, eventually made its way to developing statistical modeling methodologies for various text processing and information extraction activities.

The first attempt at using statistical methods for language understanding was the development by Scott Miller and Rich Schwartz of the Hidden Understanding Model (HUM).[93] This was actually the PhD thesis for Miller, who was enrolled at Northeastern University but did his work at BBN under a collaborative arrangement between BBN and Northeastern.[94] Since then, statistical methods have played an important role in many of our text processing activities, under the leadership of Ralph Weischedel.[95]

The ARPA SLS program evolved into what came to be known as the ATIS program, in which spoken language systems were developed for querying by voice an Airline Travel Information System (ATIS)[96] database. The accuracy of the answers that the systems gave was formally evaluated, using a previously unseen set of queries as a test set. This marked the first time that such evaluation had been done for spoken language understanding. David Stallard and Rusty Bobrow were the main developers of the BBN ATIS system.[97] The ATIS program was later followed by the ARPA Communicator program, which extended the technology to full human-machine dialogues, involving also language generation and speech synthesis. These systems were tested by users who were recruited to call the systems over the telephone and plan a given itinerary. Dave Stallard was the developer of the BBN system for this program. A simple form of voice-based interactive systems is now used commercially, such as for obtaining arrival and departure times of flights.

More recently, BBN has been working on developing speech-to-speech (S2S) translation systems, whereby two people who speak different languages can communicate with each other in a limited domain. Work in S2S translation at BBN first started in 2001 with funding from DARPA and Army Research Lab. and has continued to attract both DARPA sponsorship and private investment. Led originally by myself and Prem Natarajan, it was continued by Natarajan and later by Rohit Prasad, who now leads the S2S research under the DARPA BOLT (Broad Operational Language Translation) program, which was started in 2011. In 2010, BBN's TransTalk™ S2S system[98] was selected by the Army and DARPA for field testing and deployment.

Speech Recognition Since the 1990s

Speech recognition research activity has continued unabated at BBN to the present day, with major funding from ARPA (or DARPA),[99] as well as other agencies. A record of major achievements through the 1990s can be gleaned from the series of annual workshops that DARPA held in the area of speech and language processing during the period 1987–1999, with printed proceedings published by Morgan Kaufmann Publishers. Throughout that period, DARPA sponsored annual evaluations in speech recognition in which BBN was a top performer. The corpora used for those evaluations kept increasing in difficulty, starting with vocabularies of a few thousand words to unlimited vocabulary, and from applications like ATIS, to read speech from the Wall Street Journal, to naturally-occurring speech recorded from broadcast news.

In 2002, DARPA started the EARS (Effective, Affordable, Reusable Speech-to-text) program, which was targeted to the automatic recognition of conversational telephone speech in English, Arabic, and Chinese. For a period of six years, starting in 2005, the DARPA-funded work in speech recognition took place as part of the GALE (Global Autonomous Language Exploitation) program, which focused on the machine translation of speech or text from Arabic and Chinese into English. Since 2010, BBN has been participating in the DARPA RATS (Robust Automatic Transcription of Speech) project, whose aim is to provide automated speech activity detection, language identification, speaker identification, and keyword spotting in multiple languages under noisy communications environments. The work is being led by Spyros Matsakous, who started at BBN as a graduate student assistant in 1996 and then joined BBN full-time in 1998.

Throughout this period, BBN continued to innovate in various ways. The N-best search algorithm mentioned earlier has become a staple for much research throughout the world and its use has been extended to a number of compute-intensive problems in speech and language processing. Noteworthy innovations in speech recognition include Speaker-Adaptive Training,[100,101] Region-Dependent Transforms,[102] and Unsupervised Training.[103] In 2004, BBN was the first to demonstrate that quick transcription of speech (with a factor of ten reduction in transcription effort) was sufficient for training speech recognition models,[104] thus changing the speech transcription paradigm forever.

In an interesting project from 1999 to 2005 that was funded by NHK Broadcasting ("the BBC of Japan"), BBN helped NHK develop a real-time Japanese speech recognition system for broadcast news,[105] which was used to provide real-time, on-screen, closed captioning for the benefit of the hearing impaired. Even though the recognition system had accuracies in the high 90s, the live system included human editors who corrected the few remaining errors online. The system is still in use today.

14.6 Neural networks, OCR, and other research

In the late 1980s, ARPA started a large program to expand the theory and explore the use of artificial neural networks in various applications, including speech recognition. Rich Schwartz, Herb Gish, and I wrote the BBN proposal and won a contract to do work in that area, having succeeded in deriving some theoretical results about neural nets.[106,107] In the BBN work — which was performed largely by George Zavaliagkos and Steve Austin — neural nets were combined with HMMs to improve overall accuracy.[108] However, the use of neural nets was so computationally intensive (especially for training the neural nets) that we concluded that, while neural nets had some very good uses, speech recognition was not the best place to use them. Towards the end of the project, we convinced the ARPA program manager to allow BBN to use the remaining funds to work on using the HMM speech recognition technology to develop an optical character recognition (OCR) technology that was language independent. The result was the BBN Byblos OCR system[109] which was initially demonstrated for Arabic and has since been ported to a number of other languages.

From 1994 to 1999, the OCR work at BBN was focused on script-independent recognition of machine-printed text in Arabic, Chinese, and English. In 2000, we extended the HMM OCR approach to the recognition of text in video, also known as videotext.[110] The videotext work was funded by several agencies, including most recently by the Intelligence Advanced Research Projects Agency (IARPA), under the VACE (Video Analysis and Content Extraction) program, which ended in 2010. Starting in 2003, Prem Natarajan took over the OCR work and has since expanded it into a substantial document analysis research activity at BBN.

Some of the major milestones in recent years include the development of script-independent offline handwriting recognition technology[111] under the DARPA MADCAT

(Multilingual Document Analysis, Classification, and Translation) program; the integration of the videotext recognition technology into BBN's Multimedia Monitoring System (see next section); and the development of the operationally-deployed BBN MDATS (Multilingual Document Analysis and Translation System) which is a commercial, turnkey system for indexing large document archives. Through its many technical innovations, as well as its contributions to professional activities, BBN is now recognized as a world leader in the area of document analysis research and development.

The statistical modeling technology that started with speech recognition, and then transitioned into OCR, has recently been expanded into mainstream computer vision under Prem Natarajan's leadership. In 2010, BBN started work under the IARPA ALADDIN program, which is focused on the detection of events in video clips. BBN's top performance on NIST's annual MED (Multimedia Event Detection) evaluation held in the fall of 2011 points to a bright future for computer vision research at BBN.

The same statistical methods used in BBN's speech recognition work were also applied successfully to other areas of text processing, including topic classification,[112,113] name finding,[114] and information retrieval.[115]

A number of the speech researchers have also applied their statistical modeling expertise to the problem of machine translation.[116]

14.7 Commercial activities

Once we demonstrated real-time speech recognition on an off-the-shelf computer (described on page 366), the road to productizing the technology became open, and we got our first chance to productize in 1991. At that time, the FAA was letting contracts to develop parts of the next generation air traffic control system. As part of this, the FAA wanted to have a speech recognition system that would monitor the conversations between the air traffic controllers and the pilots and automatically update the radar screens without manual effort. The actual request for proposal came from IBM Federal Systems, which was building the overall system for the FAA. Bidders on this contract were required to have a product. Thus, BBN funded an internal R&D project to construct the HARK system, a streamlined, product version of Byblos. BBN worked on this project only long enough to deliver an initial working system, then the project stopped because the FAA cancelled the whole air traffic control program. But, as a byproduct of this effort, BBN developed the speech recognition part of a training system for air traffic controllers and licensed it to UFA Inc., which still uses it in its ATCoach product for that purpose.

In parallel, BBN received a Request for Proposal from Ford Corp. to add speech recognition to a car to allow no-hands manipulation of climate control and entertainment system functions. The main competitor was Nuance Communications, a spin-off of SRI. Mike Krasner, Robin Rohlicek, Patrick Peterson, and I prepared BBN's bid and BBN won. The work continued through much of the 1990s and the result was that the 1999 Jaguar used speech recognition technology that was developed by BBN.

Mike Krasner had joined the speech group at BBN years before, after having obtained a PhD from MIT. While he started at BBN as a researcher, Krasner saw his special knack as being more oriented toward technology management and business than toward algorithm development. In time, I began to count on Mike to help manage the department, and later turned management of the group over to him fully when I was charged with a Chief Scientist role within the company.

After the Ford contract was won, Mike Krasner took the HARK technology in 1993 into a separate commercial department independent of the government contracting environment and carried on the commercial work.

The first fielded larger-scale use of the HARK recognizer came in the summer of 1993 when the BBN Call Router became operational (and which is still in operation today — phone 617-873-8000 to try it). The call router is used to connect telephone callers to people inside an organization; one merely says the first and last names of a person and is automatically connected to that person. While there had been one or two products that could handle about 100 names at the time, BBN's call router was able to handle more than 1000 names initially and was later expanded to handle thousands of names.

In 1994, the CEO of BBN changed from Steve Levy, who had the position for many years,[117] to George Conrades. Conrades replaced some of BBN's senior divisional and business leaders who mostly had come up through the technical side of the company with people he felt would be better at marketing and business. Mike Krasner was one of the people that Conrades replaced, and Mike left BBN and started his own company in the speech area, which he later sold. The person Conrades brought in to replace Mike failed dramatically in her ability to run the commercial speech department. Jack Reilly, a capable ex-IBM marketing executive, took over the commercial speech group, and led it in a BBN-sponsored spin-off based on the call router. Needing a name for the new company, BBN discovered that it still had the name of an R&D partnership (relating to natural language understanding) that had failed.[118] This BBN spin-off is located a few miles from the BBN Technologies Cambridge site and continues to operate under the name Parlance Corp.

Also as part of the dislocation resulting from the decisions of the new CEO, Erich Bender (a long time senior manager in BBN's acoustics division) took over responsibility for what was left of BBN's commercial speech processing activities in 1996. Eventually, the new CEO's strategy for BBN led to its sale to GTE in 1997, after which GTE and Bell Atlantic merged to form Verizon.

During this era, BBN licensed its speech recognition and audio mining technologies to L&H, largely for cash (and avoided acquisition by L&H and the catastrophes that happened to all the excellent groups that L&H had acquired before its highly publicized accounting scandal and bankruptcy). Then, the speech group undertook a contract to do work for Nortel, which had disbanded its speech research group. The contract resulted in the successful fielding of an automated directory assistance system for BellSouth in the southeastern states and the development of a scalable architecture for future deployments.[119]

From 2000 to 2003, BBN's commercial speech activity was directed by Marie Meteer, who had earlier been head of speech research activities. The commercial speech group was responsible for bringing the BBN *Call Director* to market, which was the first system to combine speech recognition and statistical language-processing technologies to replace touch-tone menus many consumers must navigate when they call a business.[120] Call Director is an innovative application allowing callers to speak naturally to an open prompt: "Please tell me, briefly, the reason for your call today," and then be transferred directly to the correct agent or self serve system to solve their problem. Recipient of CallCenter Demo and Conference "Best of Show" award in 2001, BBN Hark and Call Director were first piloted in Verizon Wireless in 2000 to direct calls throughout Verizon OnLine's national footprint and in many retail call centers throughout the country. For these developments, BBN was recognized for its innovation and ability to execute in the Avios/Speechtek 2004 awards: "Best New Speech Technology: Awarded to BBN for successful large-scale commercial deployment of semantic interpretation of customer speech."

Developing commercial products that also serve the Government market has been an important and growing area of work at BBN. The various speech and language processing technologies have been integrated into a system—originally called *Rough'n'Ready* — which is able to make a rough transcription of audio into text that is ready for use by other applications.[121] The Rough'n'Ready work was led by Francis Kubala from 2000 to 2006. Capitalizing on the advances accomplished under that effort and on ongoing DARPA-sponsored research and development efforts, as well as significant privately funded efforts, in 2005 we released the BBN Broadcast Monitoring System, a commercial, turnkey system for 24x7 real-time transcription, translation, and archiving of live broadcast news videos.

In 2006, Prem Natarajan took on management responsibility for several speech-related areas, including commercial speech and language products. Since 2007, Natarajan and Amit Srivastava have led the development of several new products, such as the Audio Monitoring Component and the Multimedia Monitoring System, which handles broadcast and web sources. The new products feature a componentized architecture that offers greater integration flexibility to operational deployment teams. At the same time, we have also greatly reduced the time lag between research advances and their eventual integration into our commercial products. The advances on the engineering and core technology fronts, along with an expanded marketing and sales effort, has enabled BBN to witness substantial growth in operational deployments of its media monitoring products starting in the spring of 2007. Our focus on technological superiority and user-centered solutions design and development has resulted in BBN's products being widely deployed within US and foreign Government agencies, as well as in commercial environments.

14.8 Looking forward

The various government and commercial activities, the evolution of technology, and insights into cross-disciplinary application of BBN's approach to speech technology is paying significant dividends these days. BBN now probably has the largest government funded group in speech and language processing research in the United States. The group, numbering over 100 technical staff, has been under the leadership of Prem Natarajan since 2009.

All in all, the BBN speech group is in good shape technically and financially. I thought perhaps that I would be retired by now, but instead I find myself working day and night.

BBN has been involved in speech processing since the early 1960s. Since I joined the activity in 1970, we have moved, sometimes slowly but always surely, to an increasingly mathematical approach that today is paying significant dividends. Largely because of government funding, BBN has had a significant speech effort for many years, making smaller and larger state-of-the-art contributions in various technologies. There have been some technical firsts. There have been some novel advances. There have been some commercial successes. (There have also been some blind alleys and disappointments, but this is expected in start-of-the-art research and development — much can be learned from what doesn't work.) Anyone making a list of the top R&D places in the world for speech and language processing would have to include BBN on that list.

Acknowledgments

A number of individuals contributed substantially to this paper. Dave Walden did the research for the first section of this paper (Before 1971); he examined early BBN reports in the speech area and interviewed Ken Stevens (6/13/2003). Dave also undertook

and transcribed a long interview of me (6/14/2003) and undertook an extensive e-mail exchange with Rich Schwartz and me that provided much raw data for the main body of the paper. Dave also helped me prepare the manuscript in various more mundane ways. Rich Schwartz suggested useful changes to the initial outline for the paper and supplied much useful information. Vishu Viswanathan, now retired, provided important detail for the speech coding and speech modification sections. Pat Peterson contributed to the speech coding and commercial sections. Herb Gish provided information for the speaker recognition section. Marie Meteer and Prem Natarajan provided input to the commercial section. Prem also contributed to the parts on OCR. Other information for the paper came from Owen Kimball, Dave Stallard, Cliff Weinstein of the MIT Lincoln Laboratory, and Jerry Wolf, now retired. Drafts of the manuscript were also reviewed by many of the individuals named above. Ray Nickerson also provided a detailed review. The anonymous reviewers of the *IEEE Annals of the History of Computing*[122] pointed out specific issues and made valuable suggestions.

Dozens or hundreds of scientists, engineers and others have contributed over the years to the work described in this paper. I am pleased to acknowledge their contributions even if I did not mention them by name in this paper.

Notes and References

1. M. Mitchell Waldrop, *The Dream Machine — J. C. R. Licklider and the Revolution That Made Computing Personal*, Viking, New York, 2001.

2. John Swets, "The ABC's of BBN: From Acoustics to Behavioral Sciences to Computers," Chapter 3 of this volume.

3. Arthur L. Norberg and Judy E. O'Neill, *Transforming Computer Technology: Information Processing for the Pentagon, 1962–1986*, Johns Hopkins, Baltimore, 1996.

4. R. Bolt, F. Cooper, E. David, P. Denes, J. Pickett, and K. Stevens, "Speaker identification by speech spectrograms: A scientists' view of its reliability for legal purposes," *J. Acoust. Soc. Am.*, 47(2,Part 2):597–612, 1970.

5. Leo Beranek, "BBN's Earliest Days: Founding a Culture of Engineering Creativity," Chapter 1 of this volume.

6. Bobrow's opportunities included going with Tom Marill's start-up, Computer Corporation of America, and Ed Fredkin's start-up, Information International Inc.

7. Chapter 16.

8. BBN Report 745, K. Kryter, "Study of speech compression system (spectrum selection) interim report," April 1, 1960. A copy of this and all other BBN reports are in the BBN Library in Cambridge, MA..

9. BBN Report 1078, J. Ball, J. Coloruotolo, and K. Kryter, "Experimental Narrow-Band Speech System, Report No. 10," November 1, 1963.

10. BBN Reports 745 (citation in footnote 8), 914 (K. Stevens, K. Kryter, and M. Heckler, "An evaluation of speech compression systems," July 1, 1962), and 978 (K. Kryter, "An evaluation of speech compression techniques," February 1, 1963).

11. BBN Reports 1484 (D. Bobrow and D. Darley, "A Limited Speech Recognition system (Scientific Rept. #1)," March 1, 1967), 1667 (D. Bobrow, "A Limited Speech Recognition System," May 1, 1968), and 1819 (D. Bobrow, A. Hartley, and D. Klatt, "A Limited Speech Recognition, Final Report, System II," April 1, 1969).

12. Lucy Darley and Alice Hartley are also each listed as co-author of one of the reports.

13. BBN Report 1015, W. Teitelman, "New Methods for Real-Time Recognition of Hand Drawn Characters," June 1, 1963.

14. BBN Report 1211, J. Carbonell, M. Grignetti, and K. Stevens, "Speaker Authentication Techniques, Semi-Annual Report, 1 Jun 1964 to 1 December 1964," March 1, 1965.

15. BBN Report 1529, K. Stevens and G. vonBismarck, "A Nineteen-Channel Filter Bank Spectrum Analyzer for a Speech Recognition System, Scientific Report #2," July 1,1967.

16. Chapter 15.

17. I was officially part of the Cognitive Information Processing Group at MIT and Murray Eden was my thesis advisor. However, I did most of my thesis work in the Speech Communications Group under Stevens' direction.

18. Another student of Ken Stevens who joined BBN is Ray Tomlinson of @-sign fame, whose contributions are described elsewhere in this book. Tomlinson did his M.S. thesis under Stevens' supervision.

19. Allen Newell *et al.*, *Speech-Understanding Systems: Final Reports of a Study Group*, North-Holland, New York, 1971.

20. Computer Science and Telecommunications Board, National Research Council, *Funding a Revolution — Government Support for Computing Research*, National Academic Press, Washington, D.C., 1999.

21. The SUR program and the results of each of the contracts are documented by Dennis Klatt in *Trends in Speech Recognition*, Prentice-Hall, Inc., Wayne A. Lea, Editor, Englewood Cliffs, NJ, 1980. Part III (pages 247–421) of this compendium is entitled "Impact of the ARPA SUR Project." Part III contains seven papers: "Overview of the ARPA Speech Understanding Project," Dennis H. Klatt, pp. 249–271; "The SDC Speech Understanding System," Richard Gillman and Iris Kameny, pp. 272–293; "SRI Research on Speech Understanding," Donald E. Walker, pp. 294–315; "The HWIM Speech Understanding System," Jared J. Wolf and William A. Woods, pp. 316–339; "The Harpy Speech Understanding System," Bruce Lowerre and Raj Reddy, pp. 340–360; "The Hearsay-II Speech Understanding System: A Tutorial," Lee D. Erman and Victor R. Lesser, pp. 361–381; "Specific Contributions of the ARPA SUR Project," Wayne A. Lea and June E. Shoup, pp. 382–421.

22. And he has been more or less involved in the algorithms of most of BBN's speech projects in the years since.

23. Ibid, Wolf and Woods, pp. 316–339

24. For more detail on what happened on the natural language side of the system, see the paper listed in footnote 16.

25. The *travel task* was to recognize questions and statements from a hypothetical air traveler as if the speech understanding system was a travel agent. The travel task has been used frequently over the years by researchers wishing to test or demonstrate their systems. This task has the advantage of having a reasonably large but constrained vocabulary in a relatively narrow world of discourse, addressing a problem most observers are familiar with and can see the difficulty and value of its solution by computer.

26. J. Makhoul and J. Wolf, "Linear Prediction and the Spectral Analysis of Speech," BBN Report No. 2304, Aug. 1972.

27. J. Makhoul, "Linear prediction: A tutorial review," invited paper, *Proc. IEEE*, 63(4):561–580, April 1975. This paper was named a "Citation Classic" by the Institute for Scientific Information in the March 22, 1982, issue of *Current Contents*.

28. The word is an anagram of "spectrum" first introduced in a paper by B. P. Bogert, M. J. R. Healy, and J. W. Tukey, "The quefrency alanysis of time series for echoes: Cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking," *Proceedings of the Symposium on Time Series Analysis*, John Wiley and Sons, 1963, pp. 209–243.

29. S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics Speech and Signal Processing*, 28:357–366, 1980.

30. The branching factor is a measure of the average number of word choices at each point in an utterance. The term has since been replaced by the probabilistic measure of "perplexity."[70]

31. Klatt's paper, in the Lea 1980 collection, p. 254.

32. Chapter 17.

33. Chapter 9.

34. V. Viswanathan, J. Makhoul, R. Schwartz, and A. W. F. Huggins, "Variable Frame Rate Transmission: A Review of the Methodology and Applications to Narrowband LPC Speech Coding," *IEEE Trans. Communications*, COM-30(4):674–686, April 1982.

35. C. J. Weinstein and J. W. Forgie, "Experience with speech communication in packet networks," *IEEE J. Selected Areas in Commun.*, SAC-1(6):963–980, December 1983.

36. William Lidinsky and David Vlack, editors, *Perspectives on Packetized Voice and Data Communication*, IEEE Press, 1990.

37. Personal communication from Cliff Weinstein at Lincoln Laboratory.

38. As part of the ARPA-sponsored 1970s multi-contractor, network speech project, many issues were discovered that fed into follow-on work in another group at BBN and collaborating groups elsewhere and became the basis of the Internet Engineering Task Force effort that is the basis of Voice-over-IP technology today.

39. The speech compression algorithm used in that demo was BBN's.

40. One notable historical comment here is that Texas Instruments, in developing their innovative and very popular 1978 Speak & Spell toy, made extensive use of speech compression technology for coding the stored speech — much of it obtained through reading of various publications, including the 1974 BBN Report No. 2976 on the ARPA speech compression work, and through personal communication with BBN and other scientists.

41. J. Makhoul, V. Viswanathan, L. Cosell, and W. Russell, "Natural Communication with Computers: Speech Compression Research at BBN," BBN Report No. 2976, Vol. II, Bolt Beranek and Newman Inc., Cambridge, MA, Dec. 1974.

42. E. Blackman, R. Viswanathan, W. Russell, and J. Makhoul, "Narrowband LPC speech transmission over noisy channels," *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Washington, DC, pp. 60–63, April 1979.

43. V. Viswanathan, A. L. Higgins, and W. H. Russell, "Design of a Robust Base-band LPC Coder for Speech Transmission Over 9.6 kbit/s Noisy Channels," *IEEE Trans. Commun.*, COM-30(4):663–673, April 1982.

44. J. Makhoul and M. Berouti, "Predictive and Residual Encoding of Speech," *J. Acoust. Soc. Amer.*, 66:1633–1641, December 1979.

45. R. Viswanathan, W. Russell, and A. Higgins, "Noisy-channel performance of 16 kb/s APC coders," *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, GA, pp. 615–618, April 1981.

46. J. Wolf and K. Field, "Real-time speech coder implementation on an array processor," *IEEE Trans. Communications*, COM-30(4):615–620, April 1982.

47. V. Viswanathan, J. Tiao, L. Cosell, K. Field, A. Derr, W. Russell, and M. Krasner, "High quality 16 kbit/s voice A/D implementation," BBN Final Report No. 6205, Contract DCA100-84-C-004, April 1986.

48. J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding," invited paper, *Proc. IEEE*, 73(11):1551–1588, November 1985.

49. S. Roucos, R. Schwartz, and J. Makhoul, "A Segment Vocoder at 150 B/S," *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Boston, MA, pp. 61–64, April 1983.

50. The word "vocoder" is short for voice coder.

51. P. Jeanrenaud and P. Peterson, "Segment vocoder based on reconstruction with natural segments," *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Toronto, Canada, pp. 605–608, May 1991.

52. R. Schwartz, J. Klovstad, J. Makhoul, D. Klatt, and V. Zue, "Diphone Synthesis for Phonetic Vocoder," *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Washington, DC, pp. 891–894, April 1979.

53. V. Viswanathan and J. Makhoul, "Quantization Properties of Transmission Parameters in Linear Predictive Systems," *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-23(3):309–321, June 1975.

54. J. Makhoul, "Stable and Efficient Lattice Methods for Linear Prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-25, number 5, pp. 423–428, October 1977. This paper was awarded the IEEE Signal Processing Society Senior Award in 1978.

55. V. Viswanathan, W. Russell, and J. Makhoul, "Objective Speech Quality Evaluation of Narrowband LPC Vocoders," *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tulsa, OK, pp. 591–594, April 1978.

56. J. Makhoul, V. Viswanathan, R. Schwartz, and A. W. F. Huggins, "A Mixed- Source Model for Speech Compression and Synthesis, *J. Acoust. Soc. Amer.*, 64(6):1577–1581, December 1978.

57. Wife of Bernie Cosell who is mentioned in several other chapters.

58. J. Makhoul, "Methods for nonlinear spectral distortion of speech signals," *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Philadelphia, PA, April 1976, pp. 87–90.

59. S. Roucos and A. M. Wilgus, "High quality time-scale modification for speech," *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Tampa, FL, pp. 493–496, 1985.

60. M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Washington DC, pp. 208–211, April 1979.

61. V. Viswanathan, K. Karnofsky, K. Stevens, and M. Alakel, "Multisensor speech input for enhanced immunity to acoustic background noise," *IEEE Int. Conf. Acoust., Speech, Signal Processing*, San Diego, CA, pp. 18A.3.1–18A.3.4, March 1984.

62. V. Viswanathan and C. Henry, "Noise-immune multisensor speech input: Formal subjective testing in operational conditions," *IEEE Int. Conf. Acoustic., Speech, Signal Processing*, Glasgow, Scotland, pp. 373–376, May 1989.

63. R. Schwartz, S. Roucos, and M. Berouti, "The Application of Probability Density Estimation to Text-Independent Speaker Identification," *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Paris, France, pp. 1649–1652, May 1982.

64. H. Gish, M. Krasner, W. Russell, and J. Wolf, "Methods and experiments for text-independent speaker recognition over telephone channels," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Tokyo, Japan, pp. 865–868, April 1986.

65. L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model of ecology," *Amer. Math. Soc. Bulletin*, Vol. 73, pp. 360–363, 1967.

66. J. K. Baker, "The Dragon System — an Overview," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. 23, pp. 24–29, Jan. 1975.

67. Baker had learned about HMMs while interning at IDA. After his PhD, Baker went to work at IBM. Later, Baker and his wife Janet founded Dragon Systems.

68. Jelinek's background was in information theory.

69. F. Jelinek, L.R. Bahl, and R.L. Mercer, "Design of a linguistic statistical decoder for the recognition of continuous speech," *IEEE Trans. Information Theory*, Vol. 21, pp. 250–256, March 1975.

70. L.R. Bahl, F. Jelinek, and R.L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, pp. 179–190, 1983.

71. S.E. Levinson, L.R. Rabiner, and M.M. Sondhi, "Speaker independent isolated digit recognition with hidden Markov models," *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Boston, MA, pp. 1049–1052, April 1983.

72. Lawrence R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, 77(2):257–285, February 1989.

73. For a lay-person's introduction to HMMs used in the speech recognition application, see the following paper: Richard Comerford, John Makhoul, and Richard Schwartz, "The Voice of the Computer is Heard in the land (and it Listens too!)," *IEEE Spectrum*, 34(12):39–47, December 1997.

74. R. Schwartz, A. Derr, A. Wilgus, and J. Makhoul, "Speaker Verification IR&D Final Report," BBN Report No. 6661, Nov. 1987.

75. R. Schwartz, Y. Chow, S. Roucos, M. Krasner, and J. Makhoul, "Improved hidden Markov modeling of phonemes for continuous speech recognition," *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, San Diego, CA, paper 35.6, March 1984.

76. M. Krasner, A.W.F. Huggins, R. Schwartz, O. Kimball, Y.L. Chow, and J. Makhoul, "Development of a Vidvox Speech Communication Aid," AVIOS Conf., Arlington, VA., September 1984.

77. See pp. 275–282 of the book listed in footnote 3.

78. Jim Baker used HMMs in his PhD thesis at CMU, so HMMs were known to CMU but they did not bid their use in their Strategic Computing bid.

79. John Makhoul and Richard Schwartz., "Ignorance modeling," in Joseph S. Perkell and Dennis H. Klatt, editors, *Invariance and Variability in Speech Processes (in honor of Ken Stevens)*, pp. 344–345, Lawrence Erlbuam Associates, Publishers, Hillsdate, NJ, 1986.

80. Y. Chow, M. Dunham, O. Kimball, M. Krasner, G. Kubala, J. Makhoul, P. Price, S. Roucos, and R. Schwartz, "BYBLOS: The BBN continuous speech recognition system," *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Dallas, TX, pp. 89–92, April 1987.

81. Byblos continues to exist as a town in modern day Lebanon, the country of the author's birth.

82. R.M. Schwartz, Y.L. Chow, O.A. Kimball, S. Roucos, M. Krasner, and J. Makhoul, "Context-dependent modeling for acoutic-phonetic recognition of continuous speech," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Tampa, FL, pp. 1205–1208, March 1985.

83. The visitors included Allen Sears (ARPA Program Mangager), Ned Neuberg (DoD), and David Pallett (NIST).

84. It was after this demonstration that Ned Neuberg famously told me that this was the first time that he could say that if you poured money into speech recognition research, it would be well worth it.

85. O. Kimball, L. Cosell, R. Schwartz, and M. Krasner, "Efficient implementation of continuous speech recognition on a large scale parallel processor," *Proc. IEEE Int. Conf. Acoustic, Speech, and Signal Processing*, Dallas, TX, pp. 852–855, April 1987.

86. R. Schwartz and Y. L. Chow, “The N-best algorithm: An efficient and exact procedure for finding the N most likely sentence hypotheses,” *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Albuquerque, NM, pp. 81–84, April 1990.

87. S. Austin, P. Peterson, P. Placeway, R. Schwartz, and J. Vandergraft, “Spoken language system using commercial hardware,” *Proc. Speech and Natural Language Workshop*, Hidden Valley, PA, Morgan Kaufmann Publishers, pp. 72–77, June 1990.

88. R. Schwartz and L. Nguyen, “Single-tree method for grammar-directed, very large vocabulary speech recognizer,” U.S. Patent No. 5,621,859, April 15, 1997.

89. L. Nguyen and R. Schwartz, “Single-tree method for grammar-directed search,” *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Phoenix, AZ, pp. 613–616, March 1999.

90. Curiously, the description of these events in the book listed in footnote 3 (p. 238) makes no mention of BBN’s demonstration of a HMM-based system.

91. Discussed in the subsection beginning on page 362.

92. J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, “Continuous hidden Markov modeling for speaker-independent word spotting,” *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 627–630, May 1989.

93. S. Miller, D. Stallard, R. Bobrow, and R. Schwartz, “A fully statistical approach to natural language interfaces,” *Proc. Association of Computational Linguistics*, Santa Cruz, CA, 1996, pp. 55–61.

94. This arrangement started in 1987 when Professor John Proakis asked me if I would teach a course in speech processing at Northeastern. I later became Adjunct Professor and was able to supervise students doing their theses at BBN. This arrangement continues to this day.

95. Chapter 15.

96. D. Dahl, M. Bates, M. Brown, K. Hunicke-Smith, D. Pallett, C. Pao, A. Rudnicky, and L. Shriberg, “Expanding the scope of the ATIS task: the ATIS-3 corpus,” *Proc. ARPA Human Language Technology Workshop*, Plainsboro, NJ, Morgan Kaufmann Publishers, March 1994, pp. 3–8.

97. M. Bates, R. Bobrow, R. Ingria, S. Peters, and D. Stallard, “Advances in BBN’s Spoken Language System,” *Proc. ARPA Spoken Language Technology Workshop*, Plainsboro, NJ, Morgan Kaufmann Publishers, March 1994, pp. 43–47.

98. R. Prasad, P. Natarajan, D. Stallard, S. Saleem, S. Ananthakrishnan, S. Tsakalidis, C. Kao, F. Choi, R. Meermeier, M. Rawls, J. Devlin, K. Krstovski, A. Challenner, “BBN TransTalk: Robust Multilingual Two-Way Speech-to-Speech Translation for Mobile Platforms,” *Computer Speech and Language: Special Issue on Speech-to-Speech Translation*, Nov. 2011.

99. The ARPA name was changed to DARPA in 1972. The name then reverted to ARPA between 1993 and 1995, after which it became known as DARPA.

100. T. Anastasakos, J. McDonough, and J. Makhoul, “Speaker Adaptive Training: A Maximum Likelihood Approach to Speaker Normalization,” *Int. Conf. Acoustics, Speech and Signal Processing*, Munich, Germany, pp. 1043–1046, April 1997.

101. S. Matsoukas and R. Schwartz, “Improved Speaker Adaptation Using Speaker Dependent Feature Projections”, *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 273–278, 2003.

102. B. Zhang, S. Matsoukas, and R. Schwartz, “Discriminatively Trained Region Dependent Feature Transforms for Speech Recognition,” *Int. Conf. Acoustics, Speech, and Signal Processing*, pp. I-313–316, May 2006.

103. S. Novotney, R. Schwartz, and J. Ma, "Unsupervised Acoustic and Language Model Training with Small Amounts of Labeled Data," *Int. Conf. Acoustics, Speech, and Signal Processing*, Taipei, China, pp. 4297–4300, April 2009.

104. O. Kimball, C-L. Kao, R. Iyer, T. Arvizo, and J. Makhoul, "Using quick transcription to improve conversational speech models," *Interspeech*, pp. 2265–2268, 2004.

105. L. Nguyen, X. Guo, R. Schwartz, and J. Makhoul, "Japanese Broadcast News Transcription System," *Int. Conf. on Spoken Language Processing (ICSLP)*, Denver, CO, pp. 1749–1752, Sep. 2002.

106. J. Makhoul, A. El-Jaroudi, and R. Schwartz, "Partitioning capabilities of two-layer neural networks," *IEEE Transactions on Signal Processing*, 39(6):1435–1440, June 1991.

107. H. Gish, "A probabilistic approach to the understanding and training of neural network classifiers," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Albuquerque, NM, pp. 1361–1364, April 1990.

108. G. Zavaliagkos, S. Austin, J. Makhoul, and R. Schwartz, "A Hybrid Continuous Speech Recognition System Using Segmental Neural Nets with Hidden Markov Models," *Pattern Recognition and Artificial Intelligence*, Vol. 7, No. 4, pp. 949–963, December 1993.

109. J. Makhoul, R. Schwartz, C. LaPre, and I. Bazzi, "A Script-Independent Methodology for Optical Character Recognition," *Pattern Recognition*, Vol. 31, pp. 1285–1294, September 1998.

110. P. Natarajan, B. Elmieh, R. Schwartz, and J. Makhoul, "Video-text OCR Using Hidden Markov Models," *Int. Conf. on Document Analysis and Recognition*, Seattle, WA, Sept. 2001.

111. P. Natarajan, S. Saleem, R. Prasad, E. MacRostie, and K. Subramanian, "Multi-lingual Offline Handwriting Recognition Using Hidden Markov Models: A Script-Independent Approach," *Springer Lecture Notes in Computer Science*, pp. 231–250, 2008.

112. R. Schwartz, T. Imai, F. Kubala, and J. Makhoul, "A Maximum Likelihood Model for Topic Classification of Broadcast News," *Eurospeech*, Rhodes, Greece, pp. V3-1455–1458, Sept. 1997.

113. S. Sista, R. Schwartz, T. Leek, and J. Makhoul, "An Algorithm for Unsupervised Topic Discovery from Broadcast News Stories," *Int. Conf. Human Language Technology*, Morgan Kaufmann Publishers, pp. 110–114, March 2002.

114. D. Bikel, S. Miller, R. Schwartz, and R. Weischedel, "Nymble: A High-Performance Learning Namefinder," *Fifth Conf. on Applied Natural Language Processing*, pp. 194–201, 1997.

115. D. Miller, T. Leek, and R. Schwartz, "A Hidden Markov Model information Retrieval System," *SIGIR-99, ACM Int. Conf. Research and Development in Information Retrieval*, Berkeley, CA, pp. 214–221, Aug. 1999.

116. Chapter 15.

117. Chapter 6.

118. Ibid.

119. P. Natarajan, R. Prasad, R. Schwartz, and J. Makhoul, "A Scalable Architecture for Directory Assistance Automation," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Tampa, FL, pp. 3273–3276, May 2002.

120. P. Natarajan, R. Prasad, B. Suhm, and D. McCarthy, "Speech Enabled Natural Language Call Routing: BBN Call Director," *Proc. Int. Conf. Spoken Language Processing*, Denver, Colorado, pp. 1161–1164, September 2002.

121. J. Makhoul, F. Kubala, T. Leek, D. Liu, Long Nguyen, Richard Schwartz, and Amit Srivastava, "Speech and language technology for audio indexing and retrieval," *Proc. IEEE*, 88(8):1338–1353, 2000.

122. See the Makhoul note on page xv.

Chapter 15

Natural-Language Understanding at BBN

Ralph Weischedel

Natural-language understanding has evolved from its earliest days at Bolt Beranek and Newman, in which scientists use an early approach to parsing, to more sophisticated techniques that enable systems to extract information from open-domain text sources to fill data bases automatically.

Author's note

I joined BBN and its natural-language R&D activities in 1984. Thus, much of what I report in this article preceded my time at BBN. Nonetheless, as a graduate student and young university professor, I was well aware of, and influenced or inspired by, certain aspects of BBN's work in natural-language understanding. In this article, I emphasize those aspects of BBN's work that have been most significant to me.

BBN, since the early 1970s, has had a substantial group working in natural-language understanding. Moreover, several significant contributions came from individuals whose focus extended beyond computational linguistics to other areas of artificial intelligence (AI), such as knowledge representation and intelligent tutoring systems. The following contributions are particularly noteworthy:

- a seminal idea of semantic networks for representing the meaning of natural language
- the Lunar system, an early end-to-end system that provided natural-language access to a relational database via procedural representations of syntax and semantics
- various enhancements to augmented transition network grammars and parsers
- structured inheritance networks and limited inference, early work in the area now called description logics
- a comprehensive theory of modeling discourse
- application of statistical learning algorithms to natural-language challenges

15.1 The 1960s: Individuals and early concepts

Three individuals in particular — J. C. R. Licklider, Daniel (Danny) Bobrow, and Ross Quillian — figured prominently in BBN's AI and natural-language work during the 1960s.

Like so many of the technical threads that have endured at BBN, the natural-language understanding thread began with Licklider in the late 1950s, who was developing his ideas for man-machine interactions[1] with their considerable emphasis on AI activities.

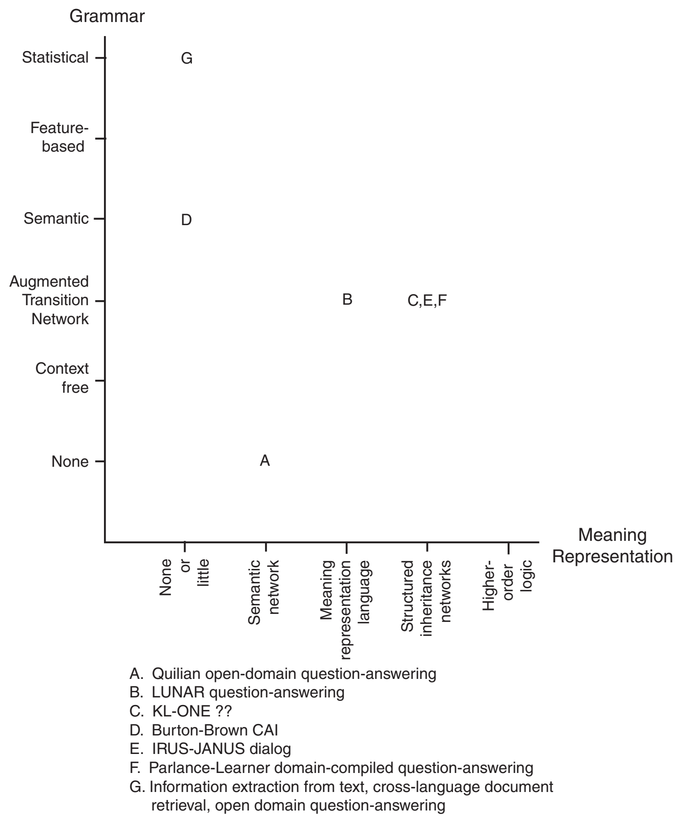


Figure 15.1. Natural-language processing applications, their grammars, and meaning representations.

Early in their days at BBN, Licklider and Tom Marill, whom Licklider recruited from Massachusetts Institute of Technology's (MIT's) Lincoln Laboratory, worked on pattern recognition projects.[2] When Licklider landed the Libraries of the Future project,[3] involving library automation, he hired MIT graduate students Danny Bobrow and Fisher Black (who later became a notable economist) part time to work on the libraries project. Their efforts touched on the area of natural-language understanding.[4,5,6]

After completing his PhD at MIT in 1964, Danny Bobrow joined BBN, having been offered the opportunity to start an AI department. Among the people Bobrow helped hire into his new AI department were Ross Quillian and Bill Woods (Bobrow had been on Woods' thesis committee).

While Bobrow spent some time working in the natural-language understanding area (and was the nominal lead person on at least some projects),[7] those individuals whom Bobrow hired were instrumental to the natural-language understanding work I describe here. From then until about 1990, BBN's natural language specialists largely resided in BBN's AI department.

Figure 15.1 depicts the various innovations, which I discuss in this article in rough chronological order, on the x- and y-axes and notes which innovations are used in which applications. The x-axis lists a series of meaning, and the y-axis lists a series of grammar and representations applied through the years. The graph summarizes major projects by grammar and meaning representations.

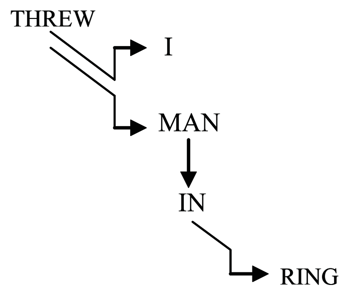


Figure 15.2. An example published by Ross Quillian. ("Semantic Memory," *Semantic Information Processing*, MIT Press, 1968, p. 262.)

Semantic networks

Ross Quillian had finished his PhD thesis at Carnegie Mellon University (CMU) in 1966. His thesis introduced semantic networks (see the x-axis of Figure 15.1) as a method of meaning representation — representing the meaning of text using a graph with labeled arcs and nodes. Figure 15.2 is an example of Quillian's that I reproduced here.[8] It is a semantic (structural) representation of "I threw the man in the ring." An upward arrow indicates the subject and downward arrows indicate grammatical objects of predicates, for example, "threw" and "in."

After joining BBN, Quillian continued to develop and disseminate his ideas. In addition to his paper on semantic networks, Quillian produced several BBN reports, including two with Alan Collins.[9] Quillian's ambition was an open-domain question-answer system (see A in Figure 15.1). Semantic networks are still in use today, although they have become more structured and more mathematically precise. Dave Walden remembers discussions with Quillian regarding his interest in the application of computers to democratic processes; in fact, after a few years at BBN, he left to join the faculty of the University of California at Irvine, where he is now a professor emeritus in political science.

15.2 The 1970s: Networks

Bill Woods, who had joined BBN's AI department in 1969, brought transition networks to BBN, led BBN's Lunar question-answering project, and was principal investigator and leader of the natural-language side of BBN's Hear What I Mean (HWIM) project.

Transition networks

For his doctoral dissertation, Woods had conceived of augmented transition networks (ATNs),[10] a concept that influenced the field of natural-language understanding for 20 years (see the y-axis of Figure 15.1).

Before Woods' work, many researchers had represented syntax using context-free grammars (see the y-axis of Figure 15.1).[11] Of course, for computer languages, context-free grammars and context-free parsing work well, for example, the following definition of arithmetic expressions involving sums of variables and numbers:[12]

```
<exp> =: (<exp> + <exp>) | (<exp> - <exp>)
<exp> =: variable | number
```

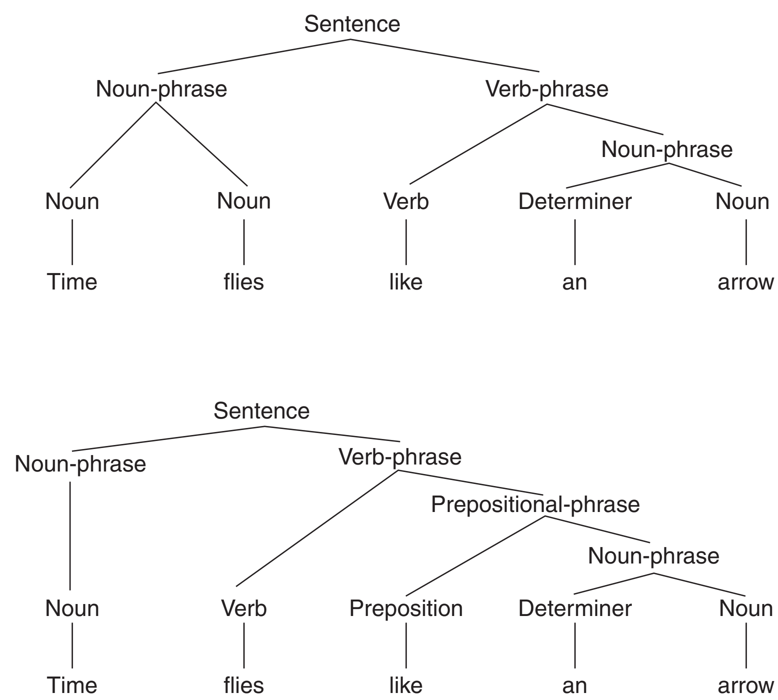


Figure 15.3 Two syntactically valid parse trees for "Time flies like an arrow".

Computer languages, designed to be unambiguous, can be defined to minimize nondeterminacy in their parsing, and context-free grammars are quite appropriate for computer languages.

Natural language is more difficult to represent. An example commonly used to illustrate the difficulty is the sentence, "Time flies like an arrow." Suppose one tries to represent this with a context-free grammar, as follows:

```
Sentence --> Noun-phrase Verb-phrase
Noun-phrase --> Noun
Noun-phrase --> Noun Noun
Noun-phase --> Determiner Noun
Verb-phrase --> Verb Noun-phrase
Verb-phrase --> Verb Prepositional-phrase
Prepositional-phrase --> Preposition Noun-phrase
```

The grammatical constraints are not strong enough in this representation; too many valid matches can result, as Figure 15.3 shows. Is the meaning of that sentence the well-known metaphor about the passage of time, or are there some little creatures known as "time flies" that, for some reason, are fond of an arrow?

Two common reasons why a formalism such as a grammar doesn't work well are that the formalism may not be strong enough, or that the language being described may be too tough to handle in the formalism. With a complex, human language such as English, context-free grammars are too limited to predict the intended interpretation, given the linguistic ambiguity. Woods enjoyed citing the example: "I saw a man in the park with a telescope." "I saw a man" is straightforward. However, "in the park" is

Figure 15.4. Lyn Bates and Rusty Bobrow working on Render Unto Syntax (RUS). (Photo courtesy of BBN Technologies.)

ambiguous — who is in the park, the man or me? There is even more ambiguity from "with a telescope": Does the man have the telescope, do I see him through a telescope, or are we talking about the park in which there is a telescope?

Woods' innovation was procedural representation of knowledge to improve the formalism and thereby enable it to better handle human language. ATNs are a way of expressing grammar with the power of a programming language built into the grammar (context-free grammars have no such power). With ATNs, you can build preferences (search strategies) into the grammar and make changes of state conditional: You can provide semantic constraints via procedure calls — in a sense, the parser is integrated with the grammar. Thus, in the example about time, you can prevent those little arrow-liking time flies from matching noun-phrase in the grammar if the system's semantics know of no such little creatures.

At BBN, Woods developed and elaborated ATNs. He was joined in this by other BBNers who advanced the state of the art of ATN use. Woods and his BBN colleagues wrote many papers relating to ATNs and used ATNs in many systems they developed, even after Woods left BBN in 1983. For instance, the Render Unto Syntax (RUS) (see Figure 15.4) and Information Retrieval Using RUS (IRUS) systems led by Lyn Bates and Robert (Rusty) Bobrow — Danny's brother — and the Janus understanding component that I led all used ATNs. I discuss these systems later.

Lunar

The Lunar system was developed primarily by Ron Kaplan, Bonnie Webber, and Woods.[13] Its goal was to support natural-language questions about rocks brought back from one of NASA's lunar landings. The Lunar system (see B in Figure 15.1) had considerable influence, showing that it was possible to develop a database access system with a natural-language front end.

Database access became a dominant application context for end-to-end system research and components for most of the 1970s and 1980s. Lunar was not the first question-answering system, and Woods was honest but still optimistic about the system's limitations. Nonetheless, Lunar had a clear design, a framework for the key challenges of language understanding, and it gave me a vision that something practical could be done with natural-language technology.

Lunar had four main components:

- It had a general morphological analyzer (that is, ability to map variations of a word such as run, ran, runs, and running onto the root, while noting the distinctions).
- It used ATNs for syntax.
- It had a meaning representation language (MRL) for representing the meaning of questions (as with ATNs, this was a procedural representation, not a declarative logic).
- It had a database of chemical data.

Examples of questions that a Lunar user might ask about rocks from a NASA landing include, for example:

> Has the mineral analcite been identified in any lunar sample?
>
> What is the average concentration of aluminum in each breccea?

HWIM (Hear What I Mean)[14]

At the beginning of the 1970s, Danny Bobrow led Bill Woods and John Makhoul to become involved with ARPA's upcoming Speech Understanding Research (SUR) program. Woods and Makhoul wrote the proposal that resulted in BBN's becoming involved in SUR (see Chapter 14 by Makhoul). Lyn Bates described this project in emails to me of 27 and 30 June 2003:

> The syntactic component of HWIM consisted of two parts, an ATN grammar of English and a parser that used the grammar to process partial utterances and to make predictions about missing words. The grammar began as a variant of that used in BBN's LUNAR system, but was extended to include more types of English constructions. The parser was completely different from the LUNAR parser, and used a mixture of top-down, bottom-up, depth first, and breadth-first strategies. Its input was not a single string of words produced by the speech recognition system, but rather a word lattice [explained below], containing a number of likely word candidates at many places in the utterance. The job of the parser was to find a path through the word lattice that constituted a meaningful utterance, possibly by filling some gaps in the word lattice as well.
>
> The ATN grammar consisted of grammatical categories (article, quantifier, adjective, noun, preposition, verb, and so on), and used a dictionary that included features that could be tested by the grammar (plural, past tense, and so on). When a large enough constituent was found (such as a noun phrase or clause), it was processed by a semantic processor to determine whether the phrase was meaningful; meaningless sequences were discarded just as ungrammatical ones were. Some parts of the parsing process were scored on an ad hoc basis, to help limit the number of alternatives that would need to be pursued.
>
> Both the grammar and the parser were written in INTERLISP.

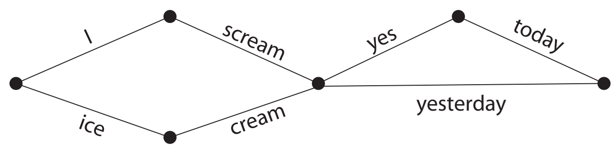


Figure 15.5 Simplified illustration of a word lattice.

Figure 15.5 illustrates a word lattice analogous to the one mentioned by Bates above. Arcs in the finite state graph represent possible words in the transcription of a speech input.

In a summer job, Bates, who had met Woods when she was in graduate school at Harvard, helped him extend and simplify his ATN grammar implemented as a Lisp interpreter. Her doctoral dissertation (*Syntactic Analysis in a Speech Understanding System*, 1975) grew out of the problem of how to organize the architecture of a speech understanding system and dealt with a variety of the problem's aspects. Consequently, it was natural for her to be involved in the SUR work at BBN.

Semantic grammars

The last work Quillian did before he left BBN involved using natural language with a computer- aided instruction system (CAI). In the late 1970s, Dick Burton and John Sealy Brown were developing computer-aided instruction systems and were not directly in BBN's natural-language understanding group. They wanted a natural language front end for their CAI systems (see D in Figure 15.1), and they invented the notion of a semantic grammar (see the y-axis of Figure 15.1).[15]

In a semantic grammar, logically enough, the grammar includes semantic information. Rather than specifying a noun-phrase, the semantic grammar might have a specific time-noun-phrase and an animal-noun-phrase; it might specify a movement-verb-phrase and a selling-verb-phrase rather than a verb-phrase. This avoids calls to a knowledge database to find out whether the noun or verb under consideration makes sense within the application domain. However, the grammar is then necessarily specific to a particular domain or application. This idea continues to be used today for applications with a narrow or limited domain.

Structured inheritance networks

Until the late 1970s, there was no mathematical semantics for semantic networks. What did the nodes and links actually mean? Bill Woods illustrated the problem[16] with a two-node, one-link semantic network in which one node is labeled telephone, the other node is labeled black, and the nodes are linked by an arc labeled color. Woods asked if the meaning of this semantic network was an instance of a black telephone, a concept consisting of all black telephones, and/or the relationship that if it's a telephone, it's black.

Woods' graduate student at Harvard, Ron Brachman, made a fundamental contribution. He proposed a new approach to semantic networks[17] called *structured inheritance networks* (see the x-axis of Figure 15.1). The new approach involved more rigorous semantics for the semantic networks themselves, structured inheritance, and class membership or class subsumption. For instance, if a dog is an animal, the dog can inherit characteristics that animals have. Separating the class hierarchy from the other binary relations among classes and defining the nodes as concepts (that is, unary terms in a logic) provided mathematical clarity to the nodes and relations of semantic networks. Additionally, the ability to state number constraints on relations also added rigor to defeasibility constraints (for example, dogs typically have four legs).

At BBN in the late 1970s, Brachman and his colleagues[18] developed the KL-ONE system (see C in Figure 15.1), which was an implementation of structured inheritance networks.[19] KL-ONE was used in several research systems, including Janus (described later).

One of the key functions needed was a form of limited inference termed subsumption: Given a Boolean combination of concepts and relations B(x), what is the term lowest in the hierarchy that subsumes B(x)? For more information, see papers from the 1981 workshop on KL-ONE, which can be accessed through the Association for Computational Linguistics' digital archive (`http://acl.ldc.upenn.edu`).

Although the ideas originated in Brachman's thesis, researchers at the University of Southern California's Information Sciences Institute (USC/ISI) and elsewhere started using those ideas — not just in language research, but also in research in knowledge representation and inference. Researchers at USC/ISI and BBN teamed in reimplementing KL-ONE and dubbed it New Implementation of KL-ONE (NIKL). Contributors included, among others, researchers Bill Mark and Norm Sondheimer at USC/ISI, and researchers Rusty Bobrow, Jim Schmolze, and Bill Woods from BBN. At BBN, an assertion mechanism and truth maintenance system were added, thanks to the integration work of Marc Vilain; this became known as KL-TWO. At USC/ISI, Robert (Bob) MacGregor took the research in new directions, resulting in the Loom knowledge representation and reasoning system; a software descendant is still in use and growing today (`http://www.isi.edu/isd/LOOM/PowerLoom/`).

Brachman's contributions had long-term impact in an area now called description logic.[20] A class of knowledge representation and reasoning systems resulted that was less powerful than first-order logic but still of considerable interest. For the next half-dozen years or so, the knowledge representation research community and the natural-language understanding community drew closer in their thinking.

15.3 The 1980s: Knowledge expands

In the 1980s, BBN's natural-language understanding group broadened to include more end-to-end natural-language engineering, as well as basic research. During the early 1980s, KL-ONE was refined and popularized. ATNs and structured inheritance networks were applied in government-sponsored projects (IRUS) and commercially oriented projects (such as Parlance — which was just a nicesounding name for the project; it was not an acronym). Additionally, Candace (Candy) Sidner of BBN, along with Barbara Grosz of Harvard, focused on discourse.

IRUS and Janus

Rusty Bobrow joined BBN as a full-time employee in 1974. He came to BBN from MIT where he had been working as resident mathematician in Jerome Lettvin's Experimental Epistemology Laboratory. At BBN, Rusty has worked as much in general AI as he has in natural language understanding. He specialized in Lisp programming and collaborated on numerous BBN projects. Lyn Bates then joined BBN in the early 1970s after completing her PhD at Harvard, where Bill Woods was her supervisor. Woods recruited her to BBN to work on the natural-language part of the SUR project (see Chapter 14), and she spent much of her time in the 1970s at the intersection of speech recognition and natural-language understanding.[21]

In about 1983, Bates and Bobrow concluded that the time was ripe to apply natural-language interfaces to database management systems.[22] Bobrow implemented an ATN compiler to optimize the speed of processing a natural-language question. Previously, ATNs had been interpreted rather than compiled. Bobrow's reimplementation compiled the ATNs into Lisp, which itself was then compiled. Bobrow was the technical lead in reimplementing the Lunar architecture to allow different applications rather than only lunar rocks. These reimplementations were also more robust than earlier systems. The grammar plus compiler were designed to be independent of the database's semantic content. The grammar and compiler as a package were named RUS (Render Unto Syntax[23]) and made available to other researchers, including myself.[24] Bates and Bobrow led a small team in building a complete end-to-end system, IRUS; secured commercial funding and created a demonstration for access to a personnel database.

Woods left BBN in 1983 to join Index Systems. Consequently, Lyn and Rusty devoted all their energy to the opportunity to create a commercial product.

IRUS was used as a component in BBN's contract as part of DARPA's Strategic Computing Program. The government was particularly interested in dialogue and in controlling multiple underlying systems, rather than just simple questioning-answering against a database. For instance, suppose a user asked the system running an airline reservations application if there were a flight from Boston to Cancun before 6 a.m. and the system answered, Yes. Next, suppose the user asked if cars could be rented at the Cancun airport, and the system answered, Yes. Finally, suppose the user were to ask, What is the cost of that flight? It is desirable for the system to have knowledge of the ongoing dialogue and to recognize that the user is referring to the Boston-Cancun flight mentioned in the next-to-last question rather than insisting that the user repeat the flight cities as part of the third query.

The version of the system used in the Strategic Computing Program was IRUS (see E in Figure 15.1). For this project, BBN worked jointly with USC/ISI. The entire system was called Janus.[25] IRUS was BBN's part; ISI's responsibility was natural-language generation, embodied in Penman, led by researchers Bill Mann[26] and Norm Sondheimer, and eventually expanded to include a single software interface to multiple underlying systems. The IRUS system could interface to multiple underlying applications, not just serve as an interface to a database. It also supported multimedia output such as tables, maps, and text. Many contributed at BBN, including Damaris Ayuso, Lance Ramshaw, Phil Resnik, and Dave Stallard.

DARPA's interest in the project included deploying the technology. Therefore, robustness, maintenance, and control of multiple underlying systems became fundamental issues. We developed ways to allow trained individuals outside of BBN to add to the system's lexicon[27] and algorithms to support access and control of multiple underlying systems.[28]

The experience exposed a gap, I believe, between the community's approach of papers plus demonstrations, and internal evaluation on test sets open to the system developers. In the 1990s, the community would adopt a paradigm from the speech community: a well-defined task specification, guidelines for human creation of an answer key, an independent (preferably automatic) scoring procedure, and a blind test set (unseen by the system developers prior to the test).

Parlance and the Learner

Roughly in parallel with the IRUS work, BBN sought outside funding to build a commercial product that would provide natural-language front ends to database management systems.[29] That version of the system was called Parlance (see F in Figure 15.1). It only interfaced to databases, but could work with a variety of databases at one customer's site or across customers.

Figure 15.6. Some participants of the Natural-Language Symposium. Left to right: Bill Woods [by then] of On Technology, Ralph Weischedel and Lyn Bates of BBN, and Bob Moore of [at that time] SRI International. (Photo courtesy of BBN Technologies.)

Bates managed this challenging effort, and Rusty Bobrow spent the mid-1980s implementing and refining this commercial system. Parlance involved an easy-to-use interface modeled on a stack of file cards. There was also a component called Learner that acquired required knowledge and vocabulary from users, thus allowing users to configure the natural-language front end to their particular databases. As with so many commercial products for natural language in that era, it is no longer available nor supported.

Discourse analysis

Sidner came to BBN in 1979 after earning her PhD at MIT. In addition to helping her natural-language colleagues carry out some bigger contracts, she began to work on the question of how to recognize intentions in discourse and relate them to the speaker's plan or method to achieve their goals, and she developed new algorithms and a prototype system. She initially focused on resolving pronoun co-reference.[30] An example Sidner considered is "John called up Mike yesterday. He wanted to discuss his homework." The challenge is to determine what the pronouns *he* and *his* refer to. Sidner later worked jointly with Barbara Grosz, of SRI and then Harvard, on a new theory of discourse structure.[31]

Natural-language symposium

As the 1980s ended, BBN hosted a Natural-Language Symposium organized by Lyn Bates and me (see Figure 15.6).[32] We recruited some of the best people available to present a horizontal slice of active research activities that, we hoped, would shape the future. A proceedings of the symposium was eventually published as a book.[33]

15.4 The 1990s: Methods

The 1990s saw natural-language methods undergo a fundamental shift to formal evaluations, empirical research, and statistical learning algorithms. By the end of 1991, a virtual revolution in methodology had begun in a handful of sites; by the end of the

1990s, the natural-language community had adopted statistical language models, other learning techniques, and corpus-based evaluation.

Changes in methods

The natural-language community underwent a change as the 1990s opened.[34] DARPA conjoined its speech research and natural language research efforts and began holding joint workshops titled Human Language Technology. DARPA, the National Institute of Standards and Technology (NIST), and the US Navy's Space and Naval Warfare Systems Center began organizing and fostering the following formal evaluations:

- Message Understanding Conferences (MUC) to evaluate information extraction from messages/documents
- Text Retrieval Conferences (TREC) to evaluate document retrieval and routing, and
- Air Travel Information System (ATIS) task for spoken-language interfaces, to allow an individual to speak to a computer assistant to make airline reservations.

At BBN, the language group merged with the speech group and began participating regularly in both MUC and ATIS.

During the 1990s, a paradigm shift occurred: The community moved from informal evaluation of algorithms and results to formal evaluations with blind test data, from relatively small test sets to corpus-based empirical research, and from purely handcrafted rule-based approaches in algorithms and applications to diverse learning algorithms. BBN's natural-language group was among the earliest to make this shift.

Statistical modeling

Another change concerned our language-understanding work that now was divided by input modality: Makhoul, Rusty Bobrow, Bates, and Stallard focused on spoken language dialogue and the ATIS task, while Damaris Ayuso, Sean Boisen, Heidi Fox, and I focused on text understanding and the MUC tasks.

In the MUC task, although the semantics of the information to be extracted is nicely circumscribed, the vocabulary and syntactic variety are not. Average sentence length in newswire documents is more than 20 words, more than double the typical utterance length of dialogues with computer interfaces. Our parsers designed for interfaces tended to time out; they encountered many words not in the handcrafted, tuned lexicon and had far more ambiguity to cope with. We felt a new approach was mandatory and were intrigued by the possibility of statistical learning approaches.

A statistical grammar (see the y-axis of Figure 15.1) is built by collecting lots of language data and annotating it. The statistics of those annotations are then embedded in the statistical models. Thus, a sentence such as "Time flies like an arrow" won't be parsed into something about little time-fly creatures that are fond of an arrow, because the language data and annotations predict as unlikely both "time flies" as a noun phrase and also "like" as a verb with object "arrow." Presumably if a sentence like this had been annotated and added to the statistical database, it would be annotated as something like:

```
[SENTENCE [NOUN-PHRASE time = NOUN]
[VERB-PHRASE flies = VERB
[PREPOSITION-PHRASE like = PREPOSITION
[NOUN-PHRASE an = ARTICLE, arrow = NOUN]]]]
```

Thus, the common meaning of the sentence would be statistically likely. This begs the question, How do you get the data to train the statistical models on the parts of speech?

Mitch Marcus of the University of Pennsylvania had the crucial insight that the key issues for using statistical models for natural-language understanding were obtaining enough training data and ensuring training data consistency. His first goal was to create 4,000,000 words of part-of-speech labeling of news. To address both the consistency and volume issues, he settled on 47 parts of speech, far more fine-grained than in a dictionary, but far fewer than in vogue in computational linguistics. With the aid of a roughly 30-page annotation manual,[35] his annotation team achieved at least 97 percent annotator consistency at roughly 2,000 words annotated per hour. BBN was one of the first users of this data.[36] Such annotation provides consistent answers in large volume to serve as the training instances for an algorithm to learn to perform such annotation.

In addition to the part-of-speech annotation, Marcus's team produced parse trees for 1,000,000 words, creating the UPenn Treebank I. (A body of parse trees for a collection of data is termed a treebank).

In parallel with our first experiments in statistical algorithms for language processing, BBN began application work, as described next (see G in Figure 15.1).

Information extraction from text

In 1991, BBN began work under a DARPA contract for information extraction from text for the purpose of automatically filling a database with information from text (the inverse of accessing a database). The goal was for the customer or user to specify the form of the database and for the language system to fill the database as it reads documents.

Damaris Ayuso had been with our group since the mid-1980s, playing a significant role in our ARPA contract under the Strategic Computing Initiative and later in a version of IRUS. Sean Boisen had joined the group in the late 1980s. Helping me with statistical modeling techniques were Scott Miller, Rich Schwartz, and Lance Ramshaw.[37]

Two interesting results came out of this period. First, the community defined an information extraction task, name tagging, where substantial success (as low as 10 percent error) has been achievable. In name tagging, the task is to identify all examples of specified types, e.g., person names, location names, organization names, dates, times, etc.

BBN's innovation here was to invent[38] a hidden Markov model for this task, [39] the first learning algorithm for name tagging that could achieve state-of-the art performance. The resulting software, IdentiFinderTM, is now used by over a dozen research laboratories in research in information extraction, information retrieval, machine translation, question answering, and summarization. At BBN, the technology has been successfully applied to languages as diverse as Arabic, Cebuano, Chinese, English, Hindi, Korean, Spanish, and Thai.

A second innovation came in adapting a lexicalized, probabilistic, context-free parsing model[40,41] to extract relations from text. In MUC-7,[42] this became the first learning algorithm to extract relations from text at an accuracy competitive with the best handcrafted, rule-based systems.

Cross-language information retrieval

In information retrieval the user types a query and the system finds the documents that seem relevant to the query. Our efforts started in 1998 when Rich Schwartz (Miller, et al., 1998) had the idea that simple statistics could be applied to the "bag of words"

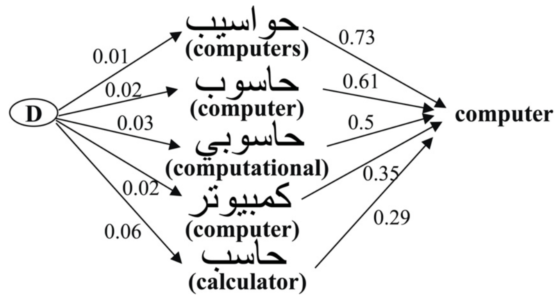


Figure 15.7. The probability that a query word "computer" came from a document D is the sum of all of the path probabilities from words in D to the word "computer".

model.[43] The results of this were very good but not as good as the best alternative method.

However, DARPA then got interested in cross-language document retrieval. In this situation, a user types the words in English and the system finds the relevant documents in a database of documents in another language, for instance, Hindi. (After retrieval, the documents can be translated to English using machine translation and/or passed to a skilled human translator.) Jinxi Xu, who received his PhD from the University of Massachusetts in 1997 and joined BBN in 1999, developed a system that extended Schwartz's idea.[44] It performed better than alternative methods of cross-language document retrieval.

The challenge with cross-lingual retrieval is to cope with two unreliable processes: bridging the language gap between the user's query and the document collection and finding relevant documents. Using a simple statistical model, we estimated the probability of a document corresponding to a term in the user's query. Figure 15.7 shows that there may be many ways of doing that. The algorithm sums the probability of all the ways a document can correspond to each term of the query.

15.5 The 2000s

As the millennium turned, BBN dug deeper into the use of statistical models and their integration with other techniques.

Question answering

Starting in 2001, BBN returned to the problem that had motivated Quillian in the late 1960s — open-domain question-answering. In the 2001 version of the problem, the open domain was over a collection of documents. The user asks questions and wants answers that can be derived from the collection of documents.

This work applies all the tools BBN has for statistical language modeling and interpretation. The user's question is reduced to a class (e.g., seeking the name of a person or seeking a biographical sketch of a person) and a set of features. The documents are modeled as a set of features, and the closest match in feature space is found. Unlike document retrieval, the features include not just words, but also semantic class structures in parse trees, relations among entities, and linguistic co-reference. The closest passages/sentences are found, and an answer extracted/synthesized from the closest passages/sentences. Roger Bock, Elizabeth Boschee, Marjorie Freedman, Nicolas Ward, Jinxi Xu, and I were key contributors. We were able to provide precise responses in English from text sources that included Arabic, Chinese, and English documents. To a query such as "Find statements made by Abu Abbas on hijacking," the system would find the relatively few statements made *by* him, not the relatively more statements made *about* him. The key to our approach was: (1) linguistic analysis of the text to find both the subject-verb-object structure of each clause in every sentence and also what the author was referring to when using a pronoun or definite description; and, (2) a flexible structure matching algorithm that finds the closest match of the semantics of the query to the semantics of the text.[45]

Machine translation

By 2003 the state of the art in statistical machine translation (SMT) involved the translation systems learning to translate from a large collection of documents and their human translations. The "training algorithm" would automatically hypothesize, sentence by sentence, the correspondence of words in the source language to words in the target language. Those correspondences provide an estimate of the probability of translation of a word or sequence of words.

At the time, statistical machine translation models learned rules that would map a word sequence from the source language to a word sequence in the target language. No explicit models of syntax or semantics were used. Nevertheless, for broadly spoken languages, it was easy to collect human translations of 200 million words (e.g. Arabic-to-English and Chinese-to-English). Thus, commercial products based on that approach started to emerge.[46]

Three developments are worth noting:

- BBN invested internal research and development funds to create its own state-of-the-art translation. Jinxi Xu and Michael Kayser were instrumental in that effort.
- BBN was among the first to investigate a syntactic model of statistical machine translation. We used a statistical parser of English to improve translation quality of the foreign source to English by scoring the syntactic quality of each hypothesized target (English) phrase. In government-run evaluations, this resulted in significantly more accurate translations than previous systems without a syntax model. By not requiring a grammar and parser of the foreign language, the technique is immediately available to many more languages than if the approach required a high accuracy parser for each source language. Libin Shen and Jinxi Xu were central in this innovation. The effort won a best paper award at a major conference in computational linguistics.[47]
- The team developed an innovative model of combining the translations of many diverse systems so effectively that translation accuracy was significantly better than any of the individual systems. System combination had been effective in prior tasks where the sequential order of the output was preserved, such as speech recognition. The challenge for machine translation was that the word order of differing system outputs could be quite different but appropriate. This effort also won a best paper award.[48]

Many other people contributed to our diverse efforts in machine translation, including senior staff such as Spyros Matsoukas and Richard Schwartz.

Human Social Cultural Behavior (HSCB)

The emphasis in natural language understanding historically has been on what is literally said; yet human communication includes many implicit meanings. By the end of the decade, BBN had begun work on identifying this type of implied meaning, such as the sentiment expressed by an author/speaker regarding a topic of interest. Furthermore, the way that an author/speaker speaks can convey a person's social intentions, such as trying to establish credibility and attempting to persuade the reader/listener. Majorie Freedman, Lance Ramshaw, and I have begun pioneering efforts in this arena.[49]

Information extraction from natural language

BBN had started research in Information Extraction in the early 90s. After 2000, two developments were significant. We developed SERIF,[50] a general language understanding engine that parsed text; mapped the parse trees to predicate-argument structure; resolved coreference; and, if given an entity-relation model or ontology, would map the text to entities and relations of that customer ontology. It was proven to be state of the art in processing English, Arabic, and Chinese in the Automatic Content Extraction (ACE) evaluations administered by the National Institute of Standards and Technology.[51] It is a core element not only of information extraction, but also of question answering, and of HSCB.

Second, Scott Miller led an effort that dramatically reduced the amount of human effort required to train name extraction.[52] Automatically induced word classes extended the effectiveness of manually marked-up examples to many more instances not seen in training. Active learning (asking individuals to judge output that is most ambiguous to the statistical model) was a second factor. Together, these two enabled BBN's name extraction system IdentiFinder™ to achieve an acceptable level of performance with as little as 10 percent of the manual effort of previous statistical approaches.

Third, Liz Boschee, Marjorie Freedman, Ryan Gabbard, and Vasin Punyakanuk developed a high performance algorithm to learn to extract binary relations, such as invent(x,y) from a few seed relation pairs, such as (Alexander Graham Bell, telephone), (Benjamin Franklin, bifocals), and (Benjamin Franklin, lightning rod). A key to success was learning patterns that depend on the predicate-argument structure in text (not just the sequence of words) and employing coreference to find examples, such as "He was awarded a patent for the telephone."[53]

Semantic resources

A limitation of statistical models has been the availability of general purpose resources marked with semantic annotation. Weischedel, Ramshaw, Mitch Marcus (University of Pennsylvania), Martha Palmer (University of Colorado), and Eduard Hovy (USC Information Sciences Institute) founded the OntoNotes effort.[54] Through the work of many, including Sameer Pradhan of BBN and Bert Xue of Brandeis University, OntoNotes created training data for parsing, semantic role labeling, and coreference in English, Arabic, and Chinese for four genres: newswire, broadcast news, blogs, and broadcast conversations (talk shows). This resource is available through the Linguistic Data Consortium and is aiding language research in many institutions.

15.6 Looking forward

Several prominent, long term members of BBN's natural language understanding group left BBN between the mid-1980s and early 1990s, e.g., Bill Woods, Ron Brachman, Candy Sidner, Bonnie Webber, and Lyn Bates. Some younger researchers became key to BBN's natural language efforts, e.g., Elizabeth Boschee, Marjorie Freedman, Scott Miller and Jinxi Xu.

Many challenges remain:

- The technologies have been broadly tested and proven on newswire, blogs, automatically transcribed news and talk shows; however, significant work remains to apply these technologies with equally accurate output on the informal language of social media, e.g. email, Tweets, and conversations.
- There are more extensive and richer resources available for English than for any other language. Applying the technologies to languages with much fewer resources is a significant research challenge.
- Deeper understanding of language is still a significant challenge. Key topics are corefence of pronouns (*it*) and descriptions (*the challenge*), event structure such as the temporal order of events and cause-effect relations in a narrative, and applying inference (when x hires y, x did not employ y immediately before, and does employ y immediately after).

Natural language processing has many potential applications, such as

- translating foreign-language documents on the Web
- automatically routing questions to an appropriate expert at a help/service telephone number
- fully automatic question answering;
- delivering answers to a Web query, as opposed to delivering pointers to Web pages
- automatically filling a structured database with desired information from text or speech sources.

Over the past 40 years, BBN natural-language understanding specialists have invented or had early involvement with a number of innovations that other researchers adopted, at least for a period of time. The group has always been motivated to try new things, in research and in applications. I am also pleased to see our R&D activities (and those of groups elsewhere) increasingly making use of controlled experiments, for that is the path to knowing what actually works and what doesn't — thus, it is also the efficient path to real long-term progress in the field.

Acknowledgments

Many contributors to BBN's research and development in natural language, as well as additional related projects, could not be named due to space constraints. I acknowledge all of these people and this work. Explicitly or implicitly, Lyn Bates, Danny Bobrow, Rusty Bobrow, and John Makhoul contributed stories and facts to this article. I especially appreciate the help of the special issue editors: Dave Walden searched BBN's archives for content for the 1960s section and helped with figures and references; Ray Nickerson provided a detailed review of the manuscript. Many thanks to Diane Bass, who originally typed this article.

Notes and References

1. M. Mitchell Waldrop, *The Dream Machine: J. C. R. Licklider and the Revolution That Made Computing Personal*, Viking Press, 2001; see also chapters 1 and 3 of this book.

2. T. Marill, “Statistical Recognition Function and the Design of Pattern Recognizers,” BBN Report 847, 1 February 1961.

3. Discussed extensively in Chapter 3.

4. D. Bobrow, Syntactic Analysis of English by Computer — A Survey, BBN Report 1055, 8 August 1963.

5. F. Black, A Question-Answering System: QAS-5, BBN Report 1063, 1 October 1963.

6. Bobrow’s PhD research, with its Student system, also had a natural-language component; see D. G. Bobrow, “Natural Language Input for a Computer Problem-Solving System,” *Semantic Information Processing*, M. Minsky, ed., MIT Press, 1968, pp. 133-215.

7. D. Bobrow, J. Fraser, and M. R. Quillian, Survey of Automated Language Processing, BBN Report 1490, 1 April 1967; D. Bobrow and J. Fraser, A Phonological Rule Tester, BBN Report 1589, 1 January 1968.

8. M. R. Quillian, “Semantic Memory,” *Semantic Information Processing*, M. Minsky, ed., MIT Press, 1968, p. 262.

9. A. M. Collins, and M. R. Quillian, “Retrieval Time From Semantic Memory,” *J. Verbal Learning and Verbal Behavior*, vol. 8, 1969, pp. 240-247; A. M. Collins and M. R. Quillian, “How to Make a Language User,” *Organization and Memory*, E. Tulving and W. Donaldson, eds., Academic Press, 1972, pp. 310-354.

10. W. A. Woods, “Transition Network Grammars for Natural Language Analysis,” *Comm. ACM*, vol. 3, no. 10, 1970, pp. 591-606; reprinted in *Readings in Natural Language Processing*, B. J. Grosz, K. Sparck-Jones, and B. L. Webber, eds., Morgan Kaufmann, 1986, pp. 71-88.

11. J. Earley, “An Efficient Context-Free Parsing Algorithm,” *Comm. ACM*, vol. 13, no. 2, 1970, pp. 94-102.

12. This is the Backus-Naur notation that computer language designers frequently use to describe their languages. The first line defines an expression to be the sum of two expressions in parentheses or the difference of two expressions in parentheses, and the second line defines an expression at its most basic to be a variable or a number.

13. W. A. Woods, R. M. Kaplan, and B. Nash-Webber, The Lunar Sciences Natural Language Information System: Final Report, 1 June 1972; W. A. Woods, “Semantics and Quantification in Natural Language Question Answering,” *Advances in Computers*, vol. 17, M. Yovits, ed., Academic Press, 1978, pp. 2-64.

14. J. J. Wolf and W. A. Woods, “The HWIM Speech Understanding System,” *Trends in Speech Recognition*, W. A. Lea, ed., Prentice Hall, 1980, pp. 316-339.

15. R. R. Burton and J. S. Brown, “Toward a Natural-Language Capability for Computer-Aided Instruction,” *Procedure for Instructional Systems Development*, H. O’Neil, ed., Academic Press, 1979, pp. 273-313.

16. W. A. Woods, “What’s in a Link: Foundations for Semantic Networks,” *Representation and Understanding: Studies in Cognitive Science*, D. G. Bobrow and A. M. Collins, eds., Academic Press, 1975, pp. 35-82.

17. R. J. Brachman, *A Structural Paradigm for Representing Knowledge*, doctoral dissertation, Applied Mathematics Dept., Harvard Univ., 1977.

18. Rusty Bobrow was a major participant — more about him later.

19. R. J. Brachman and J. Schmolze, “An Overview of the KL-ONE Knowledge Representation System,” *Cognitive Science*, vol. 9, no. 2, 1985, 1985, pp. 171-216.

20. F. Baader et al., eds., *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge Univ. press, 2003.

21. Later, when BBN’s speech and natural-language processing groups were joined together, Bates was appointed deputy department manager for the natural language part of the combined department.

22. In time, Bates, Bobrow, and others received a patent relating to this work: L. Bates, R. Bobrow, Systems and Methods for Providing User Assistance in Retrieving Data From a Relational Database, U.S. patent 6,023,697, Patent and Trademark Office, 2000.

23. Rusty had not named his program. It was first called RUS for Render Unto Syntax by Dick Burton who was writing a paper. Burton presumably was thinking of Rusty's penchant for saying "Render unto syntax that which is syntax, and render unto semantics that which is semantics." No doubt Burton also saw the pun on Rusty's name (related during a 26 April 2004 phone conversation with Rusty Bobrow).

24. I joined BBN in 1984, leaving a faculty position at the University of Delaware, where among other things, I was working to augment ATNs with more semantics (R.M. Weischedel, "A New Semantic Computation While Parsing: Presupposition and Entailment," *Syntax and Semantics II: Presupposition*, C. Oh and D. Dineen, eds., Academic Press, 1979, pp. 155-182). I was introduced to BBN by Rusty Bobrow and hired into the AI department by the then department manager, Walter Reitman. In short order I was involved in the natural-language understanding group's efforts, particularly with regard to DARPA's Strategic Computing Program.

25. R.M. Weischedel, "A Hybrid Approach to Representation in the Janus Natural Language Processor," *Proc. 27th Ann. Meeting of the Assoc. for Computational Linguistics*, Assoc. for Computational Linguistics, 1989, pp. 193-202.

26. Bill Mann at ISI is a different person than the Bill Mann who worked at BBN and is mentioned in Chapters 4 and 21.

27. D.M. Ayuso, V. Shaked, and R.M. Weischedel, "An Environment for Acquiring Semantic Information," *Proc. 25th Ann. Meeting of the Assoc. for Computational Linguistics*, Assoc. for Computational Linguistics, 1987, pp. 32-40.

28. R. Bobrow, P. Resnik, and R. Weischedel, "Multiple Underlying Systems: Translating User Requests Into Programs to Produce Answers," *Proc. 28th Ann. Meeting of the Assoc. for Computational Linguistics*, Assoc. for Computational Linguistics, 1990, pp. 227-234.

29. See Chapter 6.

30. B.J. Grosz and C.L. Sidner,"Attention, Intentions, and the Structure of Discourse," *Computational Linguistics*, vol. 12, no. 3, 1986, pp. 175-204.

31. C. Sidner, "Focusing in the Comprehension of Definite Anaphora," *Computational Models of Discourse*, M. Brady and R. Berwick, eds., MIT Press, 1983, pp. 267-330.

32. And sponsored by BBN's Science Development Program, described in Chapter 5.

33. M. Bates and R. Weischedel, eds., *Challenges in Natural Language Processing*, Cambridge Univ. Press, 1993.

34. Also about this time, I became leader of BBN's natural-language R&D group.

35. B. Santorini, Annotation Manual for the Penn TREEBANK Project, tech. report, CIS Dept., Univ. of Pennsylvania, 1990.

36. R. Weischedel et al., "Coping with Ambiguity and Unknown Words through Probabilistic Models," *Computational Linguistics*, June 1993, pp. 359-382.

37. Miller provides another example of the benefits of BBN's ongoing connections with universities. For example, John Makhoul has, as an adjunct professor at Northeastern University, set up a path that made it relatively easy to provide research funding to graduate students at Northeastern. One such student, Scott Miller, was interested in applying statistical learning to natural-language understanding, and I provided initial research funding for him to do work in this area. He wrote his PhD thesis on statistical language models for spoken dialogue interfaces under John Makhoul and Rich Schwartz, then rejoined the research on text understanding as an employee.

38. The patent was awarded to BBN through the work of Dan Bikel, Scott Miller, and Rich Schwartz.

39. D. Bikel, R. Schwartz, and R. Weischedel, An Algorithm that Learns What's in a Name, *Machine Learning* 34, 1999, pp. 211–231.

40. Collins, Michael. (1997) "Three Generative, Lexicalised Models for Statistical Parsing." In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pp. 16-23.

41. Mike Collins spent the summer of 1996 at BBN working with Scott Miller, Rich Schwartz, and myself.

42. Fox, H., Miller, S., Ramshaw, L., and Weischedel, R. "A Novel Use of Statistical Parsing to Extract Information from Text," In *Proceedings of 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, Seattle, WA., April 29-May 4, 2000 pp.226-233.

43. In a bag-of-words model, the document is reduced to a mathematical bag of the words in a document/collection. Therefore, such a model considers the frequency of the words in the document and the frequency of the words in a collection.

44. Xu, J., Weischedel, R., and Nguyen T., "Evaluating a Probabilistic Model for Cross-lingual Information Retrieval". In *Proceedings of SIGIR*, 2001.

45. E. Boschee, M. Freedman, R. Bock, J. Graettinger, and R. Weischedel, Distillation at BBN — The AGILE System, *Handbook of Natural Language Processing and Machine Translation*, Springer, 2011.

46. See Chapter 14.

47. L. Shen, J. Xu, and R. Weischedel, String-to-Dependency Statistical Machine Translation, *Computational Linguistics*, vol. 36, no. 4, pp. 649–671, December 2010.

48. Antti-Veikko I. Rosti, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie J. Dorr, Combining Outputs from Multiple Machine Translation Systems, *Proceedings of NAACL HLT 2007*, Rochester, NY, April 2007, pp. 228–235.

49. M. Freedman, A. Baron, and R. Weischedel, Language Understanding Technology for Cross Cultural Decision Making, *Advances in Cross-Cultural Decision Making*, D. Schmorrow and D. Nicholson (eds), CRC Press, pp. 395–406, 2010.

50. L. Ramshaw, E. Boschee, M. Freedman, J. MacBride, R. Weischedel, and A. Zamanian, SERIF Language Processing — Effective Trainable Language Understanding, *Handbook of Natural Language Processing and Machine Translation*, Springer, 2011.

51. `http://www.nist.gov/speech/tests/ace/`

52. S. Miller, J. Guinness, and A. Zamanian, Name tagging with word clusters and discriminative training. *Conference on Human Language Technology of the North American chapter of the Association of Computational Linguistics*, 2004

53. M. Freedman, E. Loper, E. Boschee, and R. Weischedel, Empirical Studies in Learning to Read, *Proceedings of NAACL 2010 Workshop on Formalisms and Methodology for Learning by Reading*, pp. 61–69, June 2010.

54. R. Weischedel, E. Hovy, M. Marcus, M. Palrmer, R. Belvin, S. Pradhan, L. Ramshaw, and N. Xue, OntoNotes: A Large Training Corpus for Enhanced Processing, *Handbook of Natural Language Processing and Machine Translation*, Springer, 2011.

Chapter 16

Artificial Intelligence (AI) at BBN

Compiled by David Walden

Artificial intelligence (AI) research and development has a long history at BBN, starting in the 1950s and continuing to this day. Parts of the story that pertain to speech and natural language are told primarily in the papers by John Makhoul and Ralph Weischedel in this volume (Chapters 14 and 15). But there is more to tell. The purpose of this chapter is to fill in some pieces that are not in the speech and natural language chapters and to provide a view of the larger context in which that and other AI work occurred.

Since none of BBN's leaders in artificial intelligence over the years could take the time to author this chapter when I drafted it in 2003, I (an observer of BBN's AI work for over 30 years) pulled together what I could and what seemed interesting to me. I drew on a variety of sources, but especially e-mails from and interviews with actual participants.[1]

I started by asking Bruce Roberts to give me a layman's description of the essence of the AI approach. Roberts explained that historically there has been a spectrum of AI work, from AI people who want to reproduce humanlike results (who are sometimes unconcerned with how the results are obtained) to AI people who want to create something informed by how humans do things.[2] Today's world-champion-level chess-playing programs — using vast amounts of computing power to look at far more moves than any human can — are an example of the former viewpoint. In Roberts's view, BBN's AI people traditionally leaned somewhat toward the latter viewpoint; they typically consider:[3]

- how to structure knowledge into organized conceptual chunks and the relations among them
- how to capture behavior (versus knowledge)
- how to produce a natural, powerful interface to the user

BBN's AI work has often been involved with application projects based in groups other than the AI group and described in other chapters of this book.[4] (See Billy Salter's three points at the beginning of section 16.3 below.) Nonetheless, for approximately 40 years, BBN had an explicit AI group and a majority of the AI people resided in this group, even as they sometimes applied their techniques to projects in other groups.

Section 16.1 provides a chronological history of the AI group. Section 16.2 sketches a few of the methods commonly used by BBN's AI people. Section 16.3 gives two examples of the generally *applied* nature of BBN's AI work.

16.1 Evolution of the AI group

This section gives a chronological history of BBN's AI activities. The rough time line shown in Table 16.1 may be useful as an outline for the subsections that follow.

Table 16.1 BBN AI group management time line.

1957	Licklider and Marill bring AI to BBN — no explicit AI department	[Licklider and Marill Era]
1964	D. Bobrow hired full-time explicitly to build an AI department	[Bobrow Era]
1972	D. Bobrow leaves BBN, and Carbonell takes over AI group management	
1973	Woods takes over management of the group upon Carbonell's death	[Woods Era]
1980	Makhoul and speech people leave AI group and form their own group	
1983	Woods leaves BBN, and Reitman and Stevens manage the AI group as it gets caught up in bigger corporate changes	[Transitional Years]
1984	Walker becomes manager of AI group and changes name to Intelligent Systems	[Walker Era]
1988	Natural language people join speech people in a separate group	
1997	Walker phases out of BBN and the group struggles to find a home	
1998	Intelligent Systems people find a stable home as part of BBN's Distributed Systems and Logistics Group, which uses many AI techniques in its work	

Licklider and Marill Era

J. C. R. Licklider arrived at BBN in 1957 and was well aware of the developing area of AI, which was in the air in the 1950s. Licklider soon hired Tom Marill, who was involved (along with other people) in a variety of AI-oriented work.[5]

Wally Feurzeig thinks of Marill at BBN as running the first AI group (although MIT's AI lab already existed in some form).[6] However, the BBN "AI researchers" were involved beyond AI. Licklider and Marill also brought Ed Fredkin on board, and soon Fredkin was working with AI gurus John McCarthy and Marvin Minsky to develop a time-sharing system for BBN's PDP-1 system.[7] Licklider's Libraries of the Future project[8] had several AI-oriented components and brought Danny Bobrow and Buzz Bloom to BBN as part-time employees, but the project was fundamentally about libraries, not AI.

Buzz Bloom switched to working full-time at BBN during the summer of 1961, and then was again full-time from September 1963 through May 1965. He remembers doing the following sorts of AI work:

1. Pattern recognition and scene analysis[9]
2. Hand-drawn character recognition using multi-variate discrimination analysis
3. Extension of pattern recognition and scene analysis to include learning[10]

In May 1965 Tom Marill left BBN to start his own company, Computer Corporation of America (CCA), taking Buzz Bloom (and Bill Mann) with him.[11]

Bobrow Era

Eventually Danny Bobrow came to BBN full time. He says,

> I was hired by Lick in 1962, when I was a graduate student, to work on the library project. When Lick left to go to DARPA I continued to work there part time with Tom Marill and Jerry Elkind. I graduated in 1964 and was an assistant professor at MIT. Tom Marill, working with Buzz Bloom (a roommate of mine) and me, put in a proposal to DARPA to do some natural language work. Before it got funded, Tom decided that he wanted to create a startup, and he founded CCA (Computer Corporation of America). I was invited to join him, and also to join Ed Fredkin's new startup, Information International. I decided to look around at all the options, including exploring a position with Jerry Elkind at BBN, who was now head of a

> computer science division. He offered me the opportunity to start an AI department (there was nothing left at BBN after Tom and Buzz [and Bill Mann] went off to CCA), and I found this the most interesting opportunity, so came in to BBN as an area manager — hiring Dan Murphy, Ross Quillian, and a number of others. I worked with Jerry all along to hire interesting people, including Frank Heart, Bob Kahn, Warren Teitelman (another roommate), Bill Woods (I was on his thesis committee), etc. When Jaime Carbonell Sr.[12] wanted to make the switch into AI, I supported his move into my department to work with Ross Quillian and Allan Collins (Jaime's background was in acoustics, but he was very interested in teaching and had spent a lot of time talking to Allan Collins). There eventually got to be two or three computer divisions, and Jerry was made manager of the combine; Jerry asked me to be manager of the CS Division (around 1968 or 1969) — so I was parallel to Frank Heart. I was the pusher for LISP (working closely with Dan Murphy), and we got the SDS 940 so I could have a good LISP environment. When there seemed no good successor to the 940, I helped push the idea that we should do our own operating system (TENEX) and all the TENEX crew reported to me (though obviously did their own superior work).

Bobrow had enormous impact on BBN generally and on BBN's AI area in particular. Bobrow's own work, plus the work of his hires Ross Quillian and Bill Woods, moved BBN firmly into the natural-language-understanding area, where the company remains active today.[13] In 1971 Bobrow was instrumental in hiring John Makhoul so that BBN would have a team that could bid on ARPA's Speech Understanding Research program, and Makhoul's hire moved BBN firmly into the speech-processing area, where the company also continues to be active today.[14] Murphy, Teitelman, and others did the day-to-day work of developing computer systems to support AI work (research in its own right).[15]

However, while Bobrow personally led the AI group, in his division director's role, he had TENEX developers and operators reporting to him, and BBN's early work in distributed computing[16] also resided in Bobrow's division.

On the last day of February, 1972, Danny Bobrow left BBN for Xerox PARC, and Jaime Carbonell (Sr.) became manager of the AI department.

Woods Era

Bill Woods has provided some description on his years at BBN:[17]

> I started at BBN in September of 1969, while on leave from Harvard. During that year, Jeff Warner at NASA approached me about applying my work to a system to answer questions about the Apollo 11 moon rocks. Danny Bobrow and Jerry Elkind lobbied me heavily to do the work at BBN. Jerry said that he thought Natural Language was likely to be the most supportable area in all of Artificial Intelligence. I joined BBN full time in June of 1970, initially working on this question-answering system for NASA (subsequently known as the LUNAR system). While working at BBN, I continued to teach courses and supervise graduate students at Harvard. Ron Brachman, Lyn Bates, and Bonnie Webber were some of my thesis students at Harvard who also worked at BBN.
>
> When Danny left BBN, the role of Manager of the AI Department passed to Jaime Carbonell (whose son, also named Jaime, is now a professor at CMU). When Jaime died unexpectedly, that role passed to me. (My often-cited paper "What's in a Link," which addresses the issue of meaning in semantic networks, was first published in a volume that was dedicated to Jaime.[18]) I was elected Principal Scientist in June of 1976 and continued to manage the AI Department until I left in 1983.
>
> While I was at BBN, the AI Department ran as an informal organization with teams assembled for various proposals and projects based on expertise and interests, each with a principal investigator. For example, our proposal for the DARPA

> speech understanding project in 1971 involved combining my expertise in AI and natural language understanding with John Makhoul's expertise in speech and signal processing, and our joint project with the University of Illinois on the Center for the Study of Reading drew on many people in the AI department as well as a number from other departments. While I was department manager, the AI department grew to more than 30 people, split off the speech group as a separate department (in 1980, under the leadership of John Makhoul), and then grew to more than 30 people again. Among the people I hired at BBN who went on to become significant players in Artificial Intelligence were Ron Brachman, Ron Kaplan, Lyn Bates, Rusty Bobrow, Bertram Bruce, Phil Cohen, Bonnie Webber, Candy Sidner, Jim Schmolze, David Israel, Marc Villain, and Ray Reiter. John Makhoul and John Seely Brown led significant subgroups within my department before Makhoul and the speech people became a separate group and Brown left BBN for Xerox PARC. Shortly before I left, I was negotiating with Walter Reitman to join BBN and help me with the management of the AI Department.
>
> My perspective on Artificial Intelligence has always aimed at understanding intelligence at a level that transcends both human and machine capabilities, seeking to understand the strengths and weaknesses of various techniques, and drawing on that understanding to both build artifacts that exhibit certain kinds of intelligent behavior and build artifacts that can complement human intelligence and performance. When I ran the department, the focus was on science in order to inform engineering, with a kind of interplay between theoretical research and practical applications that I called "directed research" (a combination of basic and applied research, focused on real problems). During my tenure, the AI Department was instrumental in pioneering a number of advancements in Artificial Intelligence.

As discussed in Makhoul's and Weischedel's chapters (Chapters 14 and 15), much of the work during the Woods era was in the speech recognition and natural language understanding. Also during this era, Rusty Bobrow, Danny's brother, arrived in the AI group (in 1974), from Jerry Letvin's MIT psychology lab, and he spent much of his time in the natural-language-understanding area. Bruce Roberts arrived (in 1979), from the MIT AI lab; he has spent much of his time over the years applying AI methods to training technology, not always residing in the AI group. Bobrow and Roberts, two of the longest-serving members of the AI group, are still doing this type of work today.

Transitional Years

The early 1980s was the beginning of the dot-com era for AI; and, more generally, BBN was pushing hard (as were a significant number of BBN researchers) to turn technology developed by its R&D activities into new businesses. Commercial applications of AI seemed probable, and AI people were in great demand; BBN had to make special recruiting efforts to continue to attract AI people. In about 1983, Lyn Bates and Rusty Bobrow were trying to convince BBN to set up a commercial activity based on natural-language front ends to database management systems. Rusty Bobrow was weighing leaving BBN to go to a start-up versus the potential for Parlance (a commercial natural-language system). Bill Woods did leave to go to start-up Index Systems. Ultimately, BBN did a start-up with Parlance, and Bates and Bobrow left the AI department to form an "intra-preneurial" group.[19]

As Woods was leaving BBN, Walter Reitman was recruited and took over leadership of the AI group. Walter had personal reasons for coming to Boston, and he already knew and respected lots of BBN AI people (Rusty Bobrow remembers encouraging Reitman to come to BBN). Reitman was hired and arrived before Woods left (May–June 1983) and began to help Woods manage the department. When Woods left, there was strong pressure for Reitman to stay as AI department leader, partly to quell concerns of people in the group about whether it would continue to exist after Woods left (both editors remember assuring Reitman of BBN's desire to have him stay as AI department manager).

Of his time at BBN, Reitman remembers that people like Candy Sidner and Lyn Bates began to come into their own professionally and some interesting projects got started, such as the IRS Automated Issue Identification System (AIIS — see section 16.3 below). Reitman also remembers that there was a certain amount of grumbling in the department about the IRS project. He remembers "being amused that people who were willing to work with the CIA and the NSA would want to draw the line at cooperating with the IRS."

Al Stevens had joined BBN's psychology department in 1976 after getting his PhD at UC San Diego, where Don Norman (an associate of Allan Collins) was Stevens's thesis supervisor. In the psychology department, Stevens worked with Collins, Ken Forbus, and others, moving increasingly into the area of education and training technology. By 1980 Stevens, Collins, and colleagues were looking at interactive training systems and obtained a Navy contract for development of Steamer, an advanced computer-based interactive graphics-oriented expert system for training operation and maintenance of complex steam-propulsion power plants.[20]

The next twists and turns of the AI activities were related to BBN's larger reconfigurations of itself. In 1982 I was appointed general manager of BBN's R&D activities, partly because I was willing to take initiative to rationalize various somewhat competing or somewhat isolated R&D activities into what might be more natural configurations for exploiting BBN's technology developments in the interests of faster business growth. Early on, I moved Stevens and his people from Ray Nickerson's division to Frank Heart's division: BBN president Steve Levy spoke highly of Stevens's potential to develop from a researcher into a business man, and Stevens was seeking greater visibility and independence.[21]

Ed Walker was hired in January 1984 to to manage Al Stevens's group working on expert systems and on other work at the intersection of AI and education/training technology (such as the Steamer project), by now in Heart's division. Walker's initial relationship to the AI department, managed by Walter Reitman by this time, was somewhat arm's length because it was in Nickerson's division.

When, as part of bigger BBN reorganizations, Al Stevens was appointed, effectively, to be Reitman's boss, Reitman became concerned about his continued professional growth at BBN. At that point Paladian offered him a "professionally attractive job" as VP of technology. Reitman left BBN for Palladian in August 1984.

Walker Era

When Walter Reitman left BBN, Al Stevens briefly managed the AI group himself; then Ed Walker gradually assumed the department manager role as Stevens's group and the AI department previously under Walter Reitman were combined. Walker already knew some people in the AI group (e.g., Candy Sidner) from his previous life at MIT.[22] Later, the Prophet group[23] was merged into the entity reporting to Walker.

Ed Walker says,

> At the time I was hired, AI was a hot topic for BBN and the rest of the world. The first LISP machines were being delivered, Apple MacIntoshes and heavier duty bit-mapped graphics workstations appeared on desk tops, graphical interface software, rule based systems, robots, vision systems, speech recognition systems, and knowledge representation systems made applications like Steamer and Parlance look intelligent enough and practical enough to have business potential. The Japanese trade balance

was huge, and they were buying everything they could get their hands on. Tax law favored private investment consortia. Ed Feigenbaum (a Stanford professor and grand old man of AI) convinced DARPA that Japan's MITI AI program was out to conquer the United States by force of funding and had to be countered. Major projects at BBN, SRI, ISI, CMU and elsewhere had received substantial funding. The Naval ship Carl Vinson had been launched with fanfare on its use of computing and intelligent software.

Besides acting as Al's [Al Stevens's] assistant in keeping track of the ongoing work in his small group, I led the Designer project [called DesignNet in section 16.3 below], and helped author the Butterfly proposal.[24] Designer was internally funded. The objective was to increase the productivity of the network analysis and design group that Jeff Mayerson headed at the time. Susan Bernstein co-managed the project, and Albert Boulanger, Glenn Abrett, and Bruce Roberts produced Designer. (We did a follow-on project called NewStats to aid analysts of Network Support, and Jos deBruin of the AI department helped develop Configurer to address the problem of listing the equipment required to deploy the designs produced.)

Work on RSExplore and RSDiscover add-ons to [BBN] Software Products [product] line[25] was conducted by Bruce [Roberts], Ken Anderson, and Richard Schwartz during this time frame. Fred Seibel led a project to develop a pharmaceutical factory design expert for Merck at about this time.

My primary interaction with the [natural-language work in the] AI department was with Ralph Weischedel and the IRUS project[26] to implement a natural language interface to the FRESH expert system for managing fleet scheduling for CINCPAC. Texas Instruments was the prime contractor for FRESH.

Our performance on IRUS put BBN in position to bid successfully for the follow-on CASES project. John Flynn and Ted Kral joined BBN during this time frame, and the CASES project ran under Shelly Baron's guidance.[27]

My department became involved with the Internal Revenue Service to support its attempt to infuse modern computing techniques into IRS operations. Our first project was to conduct a yearlong work-study course for four selected IRS employees. This project was led by Billy Salter. (See AIIS in section 16.3 below.)

As a result of [the IRS] work, we undertook a project to develop an expert system to analyze individual tax returns for potential audit. Dave Davis and Cynthia Whipple led this project. At about this time we also did some small projects with the Union Bank of Switzerland and the Home Office (Scotland Yard) which led to projects conducted by Mike Krasner, then of the BBN Scotland office.

Ted Kral and I collaborated on a proposal to develop a Common Prototyping Environment in support of a multi-project program in planning and scheduling funded by DARPA and RADC. Mark Burstein and Glenn Abrett worked on this project. Richard Schantz managed the last half of the CPE project. (Somewhere in the midst of this, my department acquired responsibility for developing a parallel LISP for the Butterfly. Ken Anderson led this project.) Early on in the Common Prototyping Project, we were informed of a potential quick win demonstration project that involved sophisticated scheduling algorithms produced by the small company started by Patrick Winston [the director] of the MIT AI Lab. The idea was to combine a graphical planning interface with logistical database modeling software on a Sun workstation. An early demonstration of this system was promising. The invasion of Kuwait led us to suggest a project to produce a logistics planning capability could be completed in time to support deployment of forces for the first Persian Gulf war. The project was successful and the system proved useful. It led to a long stream of funding and projects supporting logistics and transportation planning generally referred to as the "Logistics Anchor Desk."

At about this time, I became irrelevant to the history of AI at BBN.

In the world at large, less funding was available for AI research and funding emphasis was increasingly on just making application systems "smarter" and more helpful to human users. The AI people worked together with people from several other departments to propose and build systems that had an intelligent component and also made use of other BBN technology, such as distributed systems. The AIIS system mentioned above, a so-called "expert system," was such an intelligent system. In time — both in response to this trend and because he was also given responsibility for BBN's Prophet system, in the life sciences area, which had a large installed base of day-to-day users — Walker renamed the department as the Intelligent Systems Department.

In the late 1980s, the natural-language-understanding people left the Intelligent Systems Department and joined the speech-processing people in a combined department with John Makhoul as department management and Lyn Bates as deputy department manager. This left the intelligent systems people concentrating in two areas: intelligent systems and their long-term area of training technology, both in collaboration with other parts of BBN.

In 1997 or so, Ed Walker began a transition out of BBN, actually leaving in late 1998. After Ed Walker stopped actively managing the Intelligent Systems group, the intelligent systems people joined the education and training technology group for a while. However, economic pressures worked against the chances for success of this merger. Eventually, the intelligent systems people merged with the distributed systems group and some of the longtime psychology people into what was called the Distributed Systems and Logistics Group. [By the time of the final editing of this chapter in 2010, the intelligent systems people resided in BBN's Information & Knowledge Technologies activity — see the 2010 list of BBN activities near the beginning of section 22.3 on page 558.]

16.2 AI techniques

Researchers and developers with an AI orientation find themselves using many techniques for handling aspects of AI problems, for example, various heuristic and algorithmic search methods, rule-based methods, use of frames and inheritance, creation of special languages, and various methods of learning. BBN AI projects have made use of all of these.

Like many AI researchers, those at BBN did much of their work for years in LISP. Bruce Roberts explained to me that LISP is an excellent tool with which to produce such knowledge, behavior, and user-interface components. AI people see it as far and away the most productive language for programming.[28] In particular, Roberts noted that LISP is a great language in which to write other languages, and often the AI approach takes the shape of creating an appropriate language in which to write particular applications, such as interpreters or various data structures including sets of rules. The LISP and AI communities have always been early adoptors and often inventors of new programming methodologies: object-oriented programming, frames, and so on.[29]

Over the decades, BBN's AI researchers also have been especially active in developing ways of representing knowledge, as sketched through the mid-1990s in Table 16.2.

In the years immediately after those sketched in the table, the AI group also began to find uses for genetic algorithms — after Dave Davis joined BBN and encouraged their use. Davis's two books on genetic algorithms were done while he was at BBN.[30] Genetic algorithms aren't only an AI technique, although they have been used as an AI technique (e.g., in machine learning systems). They also are part of the larger field of evolutionary computation, and they frequently are used as an optimization method (e.g., to do scheduling).

No doubt the AI-oriented people BBN have continued to develop new techniques.[31]

Table 16.2 Some knowledge representation approaches.

Name	Lead person	Example application	Early year
Semantic networks	Quillian	natural language understanding	1966
Meaning Representation Language	Woods	Lunar	1971
Structured inheritance networks	Brachman	KL-ONE	1977
KL-ONE	Brachman, Schmolze and Woods	many areas	1978
FRL	Roberts (at MIT)	??	1977
PSI-KLONE	R. Bobrow and Webber	natural language understanding	1980
KL-TWO	Valain	natural language understanding	1984
NIKL	Valain?? (with ISI)	??	??1984
KREME	Abrett and Burstein	knowledge editing	1986
OMAR	Deutsch	event driven behavior language	1993
SCORE	Deutsch	tool kit for rule/event driven behavior	??

16.3 Applying AI

Looking retrospectively in 2003 at BBN's AI work over the years, Billy Salter observed the following characteristics:

- Pure research and application to real problems had frequent interplay and interaction, with research helping applications and with real-world contact making the research better[32]

- Much of the AI work was interdisciplinary, and people often didn't care where disciplinary boundaries were set. Many computer science methods were applied powerfully in BBN's AI work — whether or not they were "really" AI didn't matter.

- There was vital interplay of AI (or computer sciences more generally) and psychology: for example, how do people think about their tasks, how do people work together; what would help people do their tasks better?

As mentioned in a endnote from an earlier page (endnote 4 on page 411), other chapters in this volume mention various systems with AI components. This section describes two AI systems developed by BBN, both displaying the three characteristics noted by Billy Salter.

AIIS

The Automatic Issue Identification System (AIIS) was commonly known within BBN as the IRS system. It was an aid to human IRS auditors, looking at tax returns and searching for characteristics that would indicate high potential for claiming increased taxes. When the system identified such characteristics, a human auditor would get involved. Billy Salter says, "The IRS system was applied AI — an expert system — using some AI methods but also some boring old computer science (lots of math, table lookups, and the like) — that did a real job in the real world." It was a rule-based system running on a Symbolics LISP computer connected to a Sun workstation running the Oracle database management system. The system was developed between about 1987 and 1992.

Billy Salter has provided a description of how this project came about:

> In the early 1980s, as the AI bubble was beginning to swell, the IRS established an "AI Lab" in its research division. They let contracts to four institutions, each

to train four IRS employees in AI. Three universities and BBN won. (I wrote the proposal and ran the project; that's why I ran the big AIIS later.) The focus at BBN was on developing prototypes of potential IRS applications; almost all of the training was geared toward the needs of those projects. One of those prototypes was for a system that would flag returns and specific line items that should be audited, written in LISP on a Symbolics machine. (Albert Boulanger and Richard Shapiro were the main coders.) When the trainees went back to the IRS, they developed an RFP for an "Automated Issue Identification System," the AIIS. ("Issues" is IRS-speak for things to be audited; an issue is not quite the same as a line item but close.) BBN won that contract.

The primary goal was pragmatic: to select good returns and line items to audit. But the system also had to be amenable to annual knowledge updates by the IRS, since both tax law and, more difficult to get a handle on, patterns of under-reporting change from year to year. Identification of issues was (and is again, but never mind) a manual process: auditors on rotation sit in front of stacks of returns and spend 3 to 10 minutes with each one. They do not have time to do calculations or to look things up, and there is naturally a fair amount of "noise" inherent in the process. (The underlying phenomenon also has an inherent stochastic component: that is, for two returns with the same numbers in every line item, one can be cheating and the other not.)

Although the auditors used such "compiled knowledge" and very little math, we took advantage of what computers are good at and used a lot of math and table lookups. Eventually, we were able to reproduce auditor choices quite accurately, even though we explicitly used very different mechanisms. This is a key point: we decided very early that we would not try to capture their knowledge their way; rather, we wanted to reproduce their behavior using the best tools we had.

We worked with five expert auditors. A central challenge was how to converge on the knowledge — rules and algorithms — to use. We quickly found that the auditors could not usefully tell us enough rules to do a reasonable job; much of their knowledge and expertise were compiled and not amenable to conscious articulation or elicitation. So we presented them with many tax returns for which sometimes they had to decide what to audit and sometimes they had to review the results of the system's choices. We would then tune the rules to produce more like what they thought the results should be. But this process did not converge, for two reasons: first, changing a rule to make the results on one return better might mess up the results on another. And second, the auditors did not identify the same line items as each other or, for that matter, as themselves at different times. . . .

We used statistical insights to address this problem of convergence. Essentially, we correlated the results of the five experts and the system with each other. The correlation of each auditor with him- or herself was the "test-retest reliability," and sets an absolute ceiling on how "accurate" the system could be. That is, if an expert correlated with him- or herself .96 of the time, the system could not possibly correlate with that expert better than .96. The correlations of the auditors with each other is the "inter-rater reliability," and sets an upper limit of how well the system can correlate with them all on average. Fortunately, the auditors' correlations with each other went from the high 80s to the mid-90s. When the system's average correlation with each expert was in the middle of the range of the correlations of the auditors with each other, we knew we were doing as well as possible. We used this method for each tax year and whenever we added new categories of returns, and it became a powerful tool in assessing and tuning the system. This innovation also showed the fusion of the theoretical and the pragmatic that, I think, gave BBN's applied AI and expert system work its practical strength.

The AIIS was used operationally for approximately 20 percent of the national tax return volume for three years, where it showed improvements in equity, consistency, and dollars assessed over the existing manual system. (The first year, we ran the returns — real live tax returns, really deciding who to audit — at BBN, if you can imagine that.) The system was eliminated in huge internal political battles around "tax system modernization," a multi-billion-dollar effort that was later abandoned in the face of criticism from the GAO and Congress.

Project participant Dave Davis said that, if the AIIS technology had been permanently fielded, "it would have had the greatest return of any expert system ever built."

DesignNet

DesignNet was developed in the contract R&D side of BBN with funding from BBN's communications product subsidiary (BBNCC) to be an "expert assistant" to human network designers who planned the geographical locations of network nodes and the communications links between the nodes. Jeff Mayersohn was the contact person for BBNCC's network design activity.

According to Bruce Roberts, there were two parts of the problem: more effectively integrating the methods the network designers used, and making the process more visual so they could see their designs as they modified them. Built on a Symbolics LISP computer, the project was very successful.

Billy Salter explains that DesignNet didn't try to tell the human user how to design a network. Rather it provided multiple algorithms and multiple approaches in any order. Some designers liked to start with big nodes and high-traffic paths, some liked to start with far-apart paths, and some liked to start in other ways. This "agnostic" approach, says Salter, was "deeply right and essential to the system's success, as was its integration of graph theory, table lookups for tariffs, and lots of other stuff that was not in any sense AI."

Roberts remembers that the system included capabilities for loading data from existing or proposed networks and combinatoric algorithms and let the designer (and helped the designer) propose designs, measure designs, and improve designs. All this was constantly displayed with geographic displays connected to tabular displays so the designer could move between the two.

Although the customers for DesignNet were network designers, one of the first real uses was in a sales situation, remembers Roberts. DesignNet was being used to aid in a rather elaborate sales presentation to a major credit card company, to which BBNCC hoped to sell a network. One of the customer's people suggested an improvement to the design BBN was proposing. The suggestion was instantly plugged into DesignNet, which showed that the suggestion would reduce performance. Immediately the customer's people said, "We want that," and DesignNet significantly contributed to the network sale.[33]

Later, Dave Davis added a genetic algorithm component (see section 16.2 of this chapter) to DesignNet, which would run overnight making little changes to an existing design that improved the network design a few percent. Bill Salter remembers that the network designer users didn't like this capability, "since it couldn't 'explain' its reasoning in any way; it just kept juggling things and evaluating. This was a valuable lesson in how systems must relate to users; the designers were typically highly trained — often PhDs . . . in math or physics — and they would not accept magic from the computer."

Acknowledgments

Ray Nickerson reviewed this chapter.

Many researchers not mentioned in this paper have been important contributors to BBN's AI work over the years. I hope that someday someone will write a more comprehensive account of BBN's AI work.

Notes and References

1. In addition to the listed references, information for this paper came from Lyn Bates (e-mail of July 16, 2003); Buzz Bloom (e-mail of March 11, 2003); Danny Bobrow (e-mail of August 8, 2003); Rusty Bobrow (e-mails of February 26 and May 1, 2003); Allan Collins (e-mail of May 2, 2004); Dave Davis (e-mail of May 3, 2004); Wally Feurzeig (phone conversation of September 4, 2002); John Makhoul (e-mail of May 1, 2004); Dave Montana (e-mails of July 14–15, 2003); Ray Nickerson (e-mail of July 21, 2003); Walter Reitman (e-mail of February 3, 2003); Bruce Roberts (interview of July 11, 2003); Billy Salter (e-mails of February 27 and July 24, 2003); Al Stevens (e-mail of March 17, 2003); Ed Walker (e-mails of July 13 and August 11, 2003); and Bill Woods (e-mails of July 16, July 28, and August 13, 2003).

2. Editor Nickerson has provided a concise, useful discussion of what AI is and what motivates it in his book on *Using Computers*: Raymond S. Nickerson, *Using Computers: Human Factors in Information Systems*, MIT Press, Cambridge, MA, 1986, pp. 275–314.

3. Later in this chapter, Bill Woods describes things somewhat differently.

4. Various examples of AI work at BBN are included in other chapters in this volume. I already mentioned the Makhoul and Weischedel chapters (Chapters 14 and 15). Other chapters explicitly describing AI work include: Chapter 3 by John Swets on "The ABCs of BBN"; Chapter 8 by Raymond Nickerson and Sanford Fidell on "Psychology at BBN from the mid 1960s"; Chapter 9 by Sheldon Baron on "Control Systems R&D at BBN"; Chapter 11 by Thomas Fortmann on "Dataprobe and ADCAP"; Chapter 12 by Paul Castleman on "Medical Applications of Computers"; Chapter 13 by Wallace Feurzeig on "Educational Technology at BBN"; Chapters 4 and 21 by David Walden on the "Early Years of Basic Computer and Software Engineering" and "Later Years of Basic Computer and Software Engineering."

5. See Chapter 3 by John Swets. See also the following:
BBN Report 778, Computational Chains, T. Marill and T. G. Evans, October 1, 1960;
BBN Report 784, Transformation Rules and Program Simplification, T. G. Evans and T. Marill, February 1, 1961;
BBN Report 847, Statistical Recognition Function and the Design of Pattern Recognizers, T. Marill and D. Green, February 1, 1961;
BBN Report 1006, Techniques of Simplification, T. Marill and D. Luckham, March 1, 1963;
BBN Report 1007, The Socratic System: A Computer System to Aid in Teaching Complex Concepts, J. Swets, W. Feurzeig, A. Harris, and T. Marill, April 1, 1963;
BBN Report 1070, Studies in Automatic Pattern Recognition, Final Report, Pt 2, T. Marill, D. M. Green, D. L. Darley, T. G. Evans, and B. H. Bloom, October 1, 1963;
BBN Report 1071, Libraries' Question-Answering Systems, T. Marill, November 1, 1963;
BBN Report 1333, CYCLOPS-2: A Computer System That Learns to See, B. H. Bloom and T. Marill, November 1, 1965.

6. Feurzeig himself was hired into Marill's department. According to John Swets (e-mail of January 10, 2012), he worked half-time under a contract from Wright-Patterson Air Force Base, a contract that Swets took over when Licklider left BBN and switched the focus from a "drill-and-practice" system to a "socratic" system (see page 282).

7. Described in Chapter 4 of this volume.

8. Described in Chapter 3 of this volume.

9. BBN Report 1070, op. cit., and T. Marill, D. M. Green, D. L. Darley, T. G. Evans, and B. H. Bloom, "CYCLOPS-1: A Second-Generation Recognition System," *Proceedings of AFIPS Fall Joint Computer Conference*, 1963, pp. 27–33.

10. BBN Report 1333, op. cit., and B. Bloom and T. Marill, "Learning and Perceptual Processes for Computers," *Annals of the New York Academy of Sciences*, January, 1966, pp. 1029–1034.

11. CCA did well-known work in large-scale database management, e.g., for directory assistance systems. Bloom says, "I conceived the hashing idea . . . sometime after the Libraries of the Future project. I remember documenting the idea in a BBN internal memo. . . . This idea was later

developed in CCA's commercial information retrieval system product, and much later the idea became known throughout the database industry as bit-map indexing. After I left CCA, I wrote a paper on this idea" (B. Bloom, "Some Techniques and Trade-Offs Affecting Large Database Retrieval Times," *Proceedings of the 24th ACM National Conference*, 1969, pp. 83–95).
Editor's note: CCA also implemented the first packet-switching network experiment that led to the ARPANET and, thus, to BBN's contract (T. Marill and L. G. Roberts, "Toward a Cooperative Network of Time-Shared Computers," *AFIPS Proceedings — Fall Joint Computer Conference*, vol. 29, 1966, pp. 425–431). Bill Mann and Hal Murray did the actual programming on the TX-2 end of the experiment at MIT Lincoln Laboratory, where I used to see them in the halls. This was the first time I met Bill Mann, unless it was when Will Crowther took me rock climbing in Quincy's quarries and Bill Mann was there.

12. Jaime Carbonell Jr. also became a noted computer science person.

13. As described in Chapter 15 of this volume.

14. As described in Chapter 14 of this volume.

15. LISP and TENEX sections of Chapter 21, of this volume.

16. See Chapter 18 of this volume.

17. Bill also summarizes his work and time at BBN in the following paper: William A. Woods, "The Right Tools: Reflections on Computation and Language," *Computational Linguistics*, volume 36, number 4, December 2010 pp. 601-630, `http://www.mitpressjournals.org/doi/pdf/10.1162/coli_a_00018`. This paper is also relevant to some of the content of Chapter 15.

18. "What's in a Link: Foundations for Semantic Networks," *Representation and Understanding: Studies in Cognitive Science*, D. Bobrow and A. Collins (eds.), Academic Press, New York, 1975. When Jaime Carbonell died, Danny Bobrow and Allan Collins organized a conference in his memory and published the *Representative and Understanding Book* with chapters contributed by the conference participants. The participants included several BBN AI people in addition to Bill Woods as well as Terry Winograd, Donald Norman, Roger Schank, and Robert Abelson.

19. As described in Chapters 6 and 15 of this volume.

20. A. L. Stevens, R. B. Roberts, L. S. Stead, K. Forbus, and C. Steinberg, Steamer: Advanced Computer Aided Instruction in Propulsion Engineering, BBN Report 4702, July 1, 1981.

21. Next, I helped consolidate BBN's network sales and delivery activities into BBN Communications Corporation (BBNCC) by being willing to move Bob Bressler's more development-oriented part of Heart's division to BBNCC. I also consolidated Heart's and Nickerson's divisions, with the goal of eliminating long-standing competition between some of Heart's people and some of Nickerson's people for the same work from the same funding agencies. This competition had been frequently commented on by some of BBN's important customers, which also sometimes played the BBN groups off against each other. I named Heart as division director and Nickerson as deputy division director.

22. Also, about this time, Stevens was moving his focus to SIMNET, a distributed simulation system for team training of military tank crews (see Chapter 20).

23. Described in Chapter 12 of this volume.

24. The Butterfly project is described in Chapter 21 of this volume.

25. Also described in Chapter 12.

26. Described in Chapter 15 of this volume.

27. See also Chapter 9 of this volume.

28. Steele and Gabriel ("The Evolution of LISP," Guy L. Steele Jr. and Richard P. Gabriel, *ACM SIGPLAN Notices*, volume 28, number 3, March 1993, pp. 231–270) describe in detail what gives LISP its power and expressiveness. A version of this paper was later published on pp. 233–308 of *History of Programming Languages — II*, Thomas J. Bergin, Jr., and Richard G. Gibson, Jr., eds, ACM Press, New York, and Addison-Wesley Publishing Company, Reading, MA, 1996.

29. In more recent years LISP was less used at BBN, according to Bruce Roberts in 2003. Customers didn't know where they could get LISP programmers to maintain systems written in LISP; and, since the bust of the AI boom, fewer and fewer companies provided production-level LISP systems. Consequently, the AI people had adopted Java as a replacement for LISP. Java is a real object-oriented language. About this Roberts said, "C++ is not really an object-oriented language and no self-respecting LISP programmer would use C++." Also, Java has many useful class libraries, particularly for the Web. (Part of the reason Java became important was its emphasis on programming for the Web just when the world needed Web-programming tools.) At least one key member of the Sun team that created Java is a world expert on what made LISP great (Guy Steele).[28] Java has a garbage collector, relieving programmers of the necessity to do explicit memory management. Machines are so fast these days that the cycle of edit, compile, and run Java is nearly as fast as the traditional cycle of edit and interpret LISP.

30. *Genetic Algorithms and Synthetic Annealing*, Lawrence Davis, ed., Morgan Kaufmann Publishers, Los Altos, CA, 1987; *Handbook of Genetic Algorithms*, Lawrence Davis, ed., Van Nostrand Reinhold, New York, 1990.

31. There is a tendency among some people to be skeptical about how often the AI world produces anything that really works. The AI people, aware of this bad rap, note that that techniques developed in the AI world (such as objects and inheritance,[29] machine learning, and evolutionary algorithms) are defined as part of AI until they are widely used by other computer people and then they are called part of something else without reference to AI.

32. Bill Woods calls this kind of interplay between theoretical research and applications "directed research — a combination of basic and applied research focused on real problems."

33. Billy Salter says that DesignNet was later ported to Sun machines running C by Ken Anderson. Anderson linked this version of DesignNet to the Web, so one could obtain DesignNet results from any browser. Salter remembers the BBN salesmen using this capability as a sales tool in a visit to a senior Treasury Department official, running DesignNet from his desktop.

Part IV
Developing Computer Technology

This part of this volume contains a series of papers on BBN's efforts in developing more or less basic computer technology:

- data networking
- distributed computing
- networked email
- SIMNET
- the later years of basic computer and software engineering

A brief final chapter sketches some of BBN's activities and changes in more recent years.

Chapter 17

Data Networking @ BBN

Steven Blumenthal, Alexander McKenzie, Craig Partridge, David Walden

The chapter sketches BBN's work in data networking since the earliest days of modern, packet-based data networking.

Many of BBN's contributions to data networking derived from the fact that BBN built the ARPANET and then maintained and operated it for the U.S. Government for 20 years. Maintaining and operating a system is an excellent way to discover how that system could have been better. ARPANET was a continuing source of inspiration, frustration, and innovation, both as a stand-alone network and then as the core of the Internet.

Indeed, so many innovations occurred at BBN in the ARPANET days that presenting them all in one chapter is impossible.[1] Furthermore, various prior books have ably surveyed much of BBN's ARPANET work.[2,3,4] Accordingly, this chapter aims to illustrate the overall picture of BBN's data networking contributions by presenting a number of key research and operational themes, with a focus on contributions after the early ARPANET days. Nonetheless, the story has to start with ARPANET.

17.1 ARPANET

In the mid-1960s Bob Kahn was a relatively new faculty member at MIT, specializing in communications theory. Jack Wozencraft of MIT suggested that Kahn might like to get some practical engineering experience, and he moved to BBN, working for Jerry Elkind. In 1967 Bob was thinking about networking; he documented his thoughts in memos that Elkind encouraged him to forward to Larry Roberts at the ARPA (Advanced Research Projects Agency of the U.S. Department of Defense)[5] Information Processing Techniques Office (IPTO). Elkind was aware that ARPA was already thinking about building the first operational packet-switching network.

Frank Heart had come to BBN from MIT Lincoln Laboratory to lead the Computer Systems Division and expand its work introducing interactive computing into the medical and life sciences. Frank had gradually been recruiting several of the hardware and software people he had worked with, and come to rely on, at Lincoln Lab, including Will Crowther, Severo Ornstein, Hawley Rising, and Dave Walden. As the possibility of an ARPA procurement of a network drew near, BBN top management put Frank in charge of the potential bid effort. Thus, Bob Kahn found himself working with Frank Heart. A little in advance of the bid, a group including Frank, Bob, Severo Ornstein, and Dave Walden began to think about the network they might propose, based on the education about packet switching they received from Bob. On August 9, 1968, BBN received a copy of ARPA's Request for Quotation (RFQ) to develop and interconnect four packet-switches to be known as Interface Message Processors (IMPs).[6]

The proposal was due September 9, 1968. Heart, Kahn, Ornstein, Walden, Rising, and Bob Jacobson worked on the proposal, and people around the company, including Jerry

Elkind, Danny Bobrow, and Joe Markowitz, helped review and improve it. Will Crowther, who was in the process of moving to BBN about the time the proposal was submitted, also reviewed it and made suggestions for improvement. BBN had never previously spent so much money writing a proposal. In the fall and early winter of 1968, BBN was invited to ARPA headquarters to defend its proposal to Larry Roberts,[7] which gave the proposal team some confidence its proposal was definitely in the running to win the procurement. In late December 1968, BBN was awarded a one-year contract to develop the IMP and to deliver a four-node network connecting the University of California at Los Angeles, SRI International,[8] the University of California at Santa Barbara, and the University of Utah.

At the start of implementation the proposal team of Heart, Kahn, Ornstein, Crowther, and Walden was augmented by Ben Barker, Bernie Cosell, and others. Reflecting on that first year, Dave Walden says,

> I am struck by the general competence of the effort and the team's certainty of successful completion. Today, decades later, people often ask me whether I was worried about being a member of a team that had so much to accomplish in only one year. Of course developing that first IMP system was a relatively small project compared to the massive extent of what people think of today when they think of the Internet. We also knew we had a tight schedule, and we worked very hard. However, I didn't see any real worry from any member of the team at any time. We were a small team of highly motivated and, on average, highly experienced people that worked well together during that first year. We were one of those "hot teams" that sometimes get written up in management books. We were very focused — the team was enormously pragmatic and concentrated on getting a system delivered on time that worked "well enough."

Bob Kahn was the one member of the team who wasn't convinced that "well enough" was adequate. He could see flaws in the routing and congestion-control implementation plans: flaws that might not have an impact at first but that he was sure would soon become serious problems. As the group's theoretician he felt it improper to deploy a system with known flaws, and as a result felt increasingly at odds with the direction of the implementation team. (He may also have felt that BBN would never have had an opportunity to bid on the network if it had not been for his memos to Roberts, and therefore that his views ought to carry more weight.)

The First Packet Switches

There were debates within ARPA about whether the network should organize itself or be centrally managed from a controlling computer. BBN felt the network should be self-organizing. Pursuing that goal led to important characteristics of the first IMPs (and the network created from them), including:[9]

- Features to minimize the need for on-site assistance and support for cross-network diagnosis, debugging, and new releases[10]
- Considerable facilities for network monitoring and measurement
- No constraints put on the data that hosts could exchange over the network
- Initial distributed algorithms for IMP-to-IMP communications and network routing
- Much less successful initial algorithms for host-to-IMP and source-IMP-to-destination-IMP communications — the former was too limited because of the assumption of a direct electrical connection rather than a remote communications interface, and

the latter was simply inadequate to the congestion control and multiplexing task it was designed for

- A design and implementation that was very high-performance in terms of use of memory and machine cycles and very reliable in terms of the IMPs' not crashing because of coding bugs.

Somewhat to the surprise of the people outside BBN, the first IMPs were completed and delivered on time.[11] Although there were some missteps, the initial IMP design and implementation was quite robust. It provided good support for the host experiments and a powerful mechanism for releasing incremental improvements as they were needed.

ARPANET Grows

After installing the first 4 ARPANET nodes in 1969, ARPA expanded the network to 19 nodes. As the network grew, it became increasingly difficult to recognize troubles and initiate the appropriate corrective action. Each IMP was programmed to keep track of its local environment (ciruit and host status, software version, machine front panel switch settings, traffic statistics, etc). The fifth ARPANET IMP was installed at BBN in early 1970; once it was in place all the IMPs sent periodic status reports to the IMP 5 "console," a Model 33 Teletypewriter (10 characters/second). Bernie Cosell recalls:[12]

> I was mostly the guy reading the TTY output at that time. But as the net grew, that was getting to be harder and harder (too damn much paper clanking away in the back room). There was a spare [Honeywell] 316 in the room next to the PDP-1, and we got some kind of fairly-high-speed printer for it.. So it [the Honeywell 316] first came up just as a network host and just printed out all the junk on the nicer printer. But I added a bunch of smarts to it. In particular, I kept track of things and only reported changes (who was up, who was down, some heuristics for when a report was overdue). Then Jon Cole cobbled up a lights box [which displayed IMP status or circuit status on a set of 32 lights] and then the 316 was modified to signal an alert and flash appropriate lights. Another thing I did was to put in "topology" code — so that if, say, IMP 5 went down, it would figure out that we had no reporting-path to IMP 6 and put it in a (quiet) limbo instead of announcing that it was down. This proved to be useful for other similar things; when a line went down and segmented the net, the code was smart enough to report "either this IMP or this line is down" and not list another 20 IMP Down reports.

Also, to support this expansion, two key people had joined the team by 1971: Alex McKenzie and John McQuillan. They both were initially involved with network operations. McQuillan worked with Cosell to implement the network monitoring software on the 316, and McKenzie began to lead the operations component of BBN's efforts.[13]

Beyond the tasks of operating and managing the ARPANET, the team known as the IMP Guys had to solve a number of important problems, including:

- Fixing the problems with the initial design for end-to-end message reassembly to deal with congestion problems[14]
- Augmenting the IMP with a terminal handling capability[15,16]
- Supporting satellite links between IMPs[17]
- Developing a multiprocessor version of the IMP[18]
- Replacing the original (and first) distance-vector routing algorithm with the first link-state routing algorithm[19]

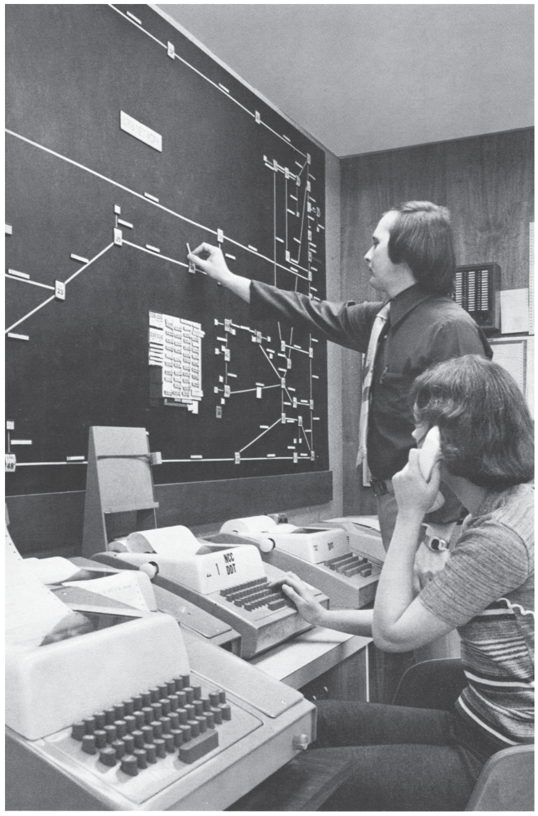


Figure 17.1. Early version of the ARPANET Network Operations Center (NOC): Jim Powers standing.

The leadership of the IMP software development effort transitioned over time from the initial team of Crowther, Cosell, and Walden; leadership first passed first to McQuillan, then to Paul Santos, next to Jim Herman, and finally to Ken Hahn. Furthermore, once the ARPANET became operational, there was a tremendous effort to develop the host-to-host protocols that ran over the network. ARPA funded a number of groups (mostly at sites that had or were anticipated to get an IMP) to study and develop protocols. BBN, as the operator of the ARPANET and also as the maintainer of the TENEX operating system (one of the major research operating systems of the time[20]), had an important role in developing or refining several early ARPANET protocols: Network Control Protocol (the predecessor to TCP),[21] Initial Connection Protocol,[22] Telnet,[23,24] File Transfer Protocol (FTP),[25] and, of course, the e-mail protocols.[26]

In October 1972, at the first International Conference on Computer Communication in Washington, D.C., the ARPANET community provided a multiday live demonstration of the technology. Larry Roberts had conceived the idea a year earlier and asked Bob Kahn to do the detailed planning. Bob asked Al Vezza of Project MAC at MIT to assist him. The two of them arranged for a ballroom at the conference hotel to be the demonstration site. BBN delivered a TIP (IMP with additional hardware and software to support 63 terminals), AT&T donated circuits to two nearby ARPANET sites, and dozens of terminal manufacturers each loaned a terminal or two. The ARPANET host sites arranged for demos of the software unique to their institutions, and instructions for accessing the demos were collected by Bob Metcalfe (a Harvard PhD candidate) in a booklet titled "Scenarios for Using the ARPANET," which was handed out to conference attendees.[27] The demonstration was a great success and convinced most people at the conference that packet switching was a viable networking technology.[28] In the months following the demo, Bob Kahn moved from BBN to ARPA to assist Roberts in expanding the scope of packet-switching experiments.

In July 1975, ARPA declared that ARPANET was an operational network and transferred management responsibility for ARPANET to the Defense Communications Agency (DCA). BBN continued to have day-to-day operational responsibility, now under contract to DCA rather than ARPA. ARPA paid a fee to DCA for each ARPANET location it sponsored; other parts of the government (for example, the Army) also paid DCA for their locations. DCA agreed to operate ARPANET until mid-1978, after which it was to be replaced by an equivalent service provided by a military network. The anticipated military network was AUTODIN II (discussed below).

ARPANET Influences and Spin-offs

In the early years following the original ARPANET IMP development and deployment, BBN and BBN people were involved in or influenced a significant number of more or less derivative networks. BBN established a small networking group in BBN's Rosslyn, Virginia, office led by Eric Wolf, who was later joined by Eric Elsam and Bob Hess. The group managed the modification and deployment of the ARPANET IMPs in various government applications, among them the COINS network.[29]

Dave Walden spent a year (September 1970 to September 1971) working for Norsk Data Elektronikk in Oslo, Norway, before returning to BBN and the IMP Guys team. There he led the development of a small ARPANET copy on Norsk Data computers.[30] Walden and Alex McKenzie consulted to Louis Pouzin's team at the French Research Establishment that developed the French Cyclades network;[31] the French team used what they learned to avoid doing the same thing (whether it had been successful or not) in the network they built, thus assuring its deliberate uniqueness. Walden also gave a paper in Japan in 1975[32] that led to at least one Japanese network's having some design elements copied from the ARPANET IMP.

BBN had a business arrangement with Logica in the United Kingdom and SESA in France whereby the Logica and SESA teams learned the details of the ARPANET IMP design and implemented networks in Europe. To support pursuit of European opportunities, Frank Heart had Peter Kirstein of University College London[33] on a yearly retainer for a number of years. Beginning in 1979, Ira Richer, Tony Michel, and Alex McKenzie each spent a year or two on site at Olivetti headquarters in Italy, helping Olivetti design and build a packet network for a consortium of Danish savings banks.

BBN was also involved as a partner with Honeywell in several proposals to build private packet networks, but Honeywell never was successful in selling any of them. However, as an indirect result of this partnership, BBN implemented an experimental network on Honeywell hardware for a large New York City bank. This network operated for several years and served as the base for several trials within the bank to use packet switching to support various bank internal operations.

The first spin-off was Packet Communications Inc. (PCI) which was started by three BBNers: Ralph Alter, Lee Talbert, and Steve Russell. Talbert had been hired by BBN to try to commercialize the technology and was dissatisfied with the slow pace of his progress within the company. Alter and Russell were engineers who were involved with the ARPANET's growth. PCI was the first packet carrier licensed by the Federal Communications Commission (FCC) but failed to raise enough funding to reach sustainable operation.

The most notable spin-off was the first operational packet-switching common carrier, Telenet, which BBN founded in 1972. The business aspects of this effort have been described by Steve Levy.[34] Telenet hired Larry Roberts to be its CEO (and Bob Kahn assumed leadership of the ARPA IPTO office). BBN sent its IMP software, and developers Steve Butterfield and Chris Newport, to Telenet in Washington, where Butterfield converted the software to run on a later-model computer. As soon as possible, Telenet redid its packet-switching software on its own hardware. In 1979 BBN arranged the sale of Telenet to GTE and used its share of the substantial return on investment to develop its own networking business.

Perhaps more interesting than the ARPANET spin-offs was the strong desire of both researchers and corporations to develop their own independent versions of packet switching. Some teams wanted proprietary protocols that they could control (for example IBM's Systems Network Architecture (SNA) and Digital Equipment Corporation's DECNET). Others simply wanted to explore alternative design choices.

AUTODIN II: ARPANET Becomes the Military's Key Network

The U.S. Department of Defense (DoD) developed AUTODIN (Automatic Digital Network) in the 1960s as a DoD-wide message-switching system. It was operated by Western Union under a contract from the Defense Communications Agency (DCA). In 1976 DCA announced that it would replace AUTODIN with a new packet-switching network called AUTODIN II. BBN was the developer and operator of the first and biggest packet-switching system, the ARPANET, and by this time ARPANET's operation was being managed by BBN under contract to DCA, giving BBN much contact with DCA people. Thus, BBN thought bidding on AUTODIN II made good sense. The company formed a team with Pace Communication and Telenet, and put in the enormous effort to bid to build AUTODIN II. The large bid package was submitted, an end-of-bid dinner was held at Joyce Chen's restaurant near BBN in Cambridge, and the fortune cookie prediction looked promising: "You will have bushels of gold."

However, in bidding for the AUTODIN II contract, the BBN team made a tremendous mistake. Rather than writing the proposal in the form requested by DCA, BBN management insisted on writing the proposal in a different format which seemed more coherent to them. DCA's format, however, reflected DCA's proposal review process, in which individual pieces of a proposal were handed to different review teams. BBN's proposal suffered in this review scheme and BBN lost the contract to a team led by Western Union.

Over the next 18 months, Western Union and its team missed some major milestones in the development schedule, and DCA began to worry that Western Union was failing. Pressure began to build for DCA to adopt the already-working ARPANET technology. As one example, Keith Uncapher, director of the University of Southern California's Information Sciences Institute (USC ISI), advised ARPA and DCA to accept the ARPANET technology. In time, DCA opened a new bid for the contract. BBN competed against Western Union for this contract. Both the BBN team and Western Union team were led by people from the government. The BBN proposal was a large document (4 inches thick) addressing all the issues that the DCA team had identified, including network design, security analyses, logistics, reliability, and vulnerability to nuclear attack. The final recommendation was made by a technical team set up by DCA.

Following the DCA technical team's recommendations, Deputy Secretary of Defense Frank Carlucci stopped the Western Union AUTODIN II contract (resulting in large cancellation payments to Western Union), and on April 2, 1982, told BBN to begin adapting the ARPANET technology to DCA's needs. Among other adaptations, DCA wanted the host interface to be the X.25 standard of the Comité Consultatif Internationale de Télégraphie et Téléphonie (CCITT).[35] This contract established ARPANET as the Defense Data Network (DDN) which supported the DoD's data communications requirements for the next 10 years.[36]

17.2 On to the Internet

ARPANET kicked off a rapid growth in network technology, including satellite networks, local area networks (LANs), and packet radio networks. The networking community realized that interconnecting these different types of networks was a serious problem, and members of the community took a variety of somewhat parallel steps to deal with it. The end result of these parallel activities created what we know today as the Internet.

Interconnecting Networks

As previously noted, Larry Roberts initiated a public demonstration of ARPANET in October 1972 at a conference in Washington, D.C. This conference was attended by dozens of people from the ARPANET community and by representatives of the National Physical Laboratory (NPL) network in the United Kingdom and the Cyclades network in France (both experimental packet networks). It was also attended by the Canadian, French, and U.K. telephone companies, all of whom were designing national packet networks. These groups, plus researchers from Japan, Norway, and Sweden, got together during the conference to discuss how to interconnect these and future packet networks so that a host attached to one could communicate with a host on any other. The group called itself the International Network Working Group (INWG) and began an immediate exchange of papers (INWG Notes).[37] INWG Note #6, distributed the next month by Donald Davies of NPL, stated "It was agreed [in October] that . . . networks will probably be different and thus gateways [routers] between networks will be required." Davies went on to set forth questions on routing, flow control, addressing, and so forth that needed to be considered in the design of the constituent networks, gateways, and host protocols. Within a few months INWG formally became a subcommittee of the International Federation for Information Processing (IFIP), which gave it standing to participate officially in international standards-making organizations.

1972 marked the beginning of a new four-year cycle in the standardization activities of the CCITT, a treaty organization of national telephone companies. During 1973 and 1974 both the CCITT and the INWG discussed various proposals for interconnections between public packet-switched networks. One significant proposal was described in a paper by Vint Cerf and Bob Kahn (by then at ARPA) proposing a specific "Internet Transmission Control Program" (TCP) for host computers and gateways, which implemented a reliable byte-stream.[38] Other submissions to INWG from France and the United Kingdom segmented the data stream into "letters" rather than bytes for error and sequence control. However, each of these proposals assumed that the individual networks could be relied on only to carry independent data packets without error correction or sequencing, an activity known as a datagram service. CCITT had not really accepted the idea of a datagram service[39] and was discussing a "virtual circuits" concept within the network; this was intended to relieve the hosts of any responsibility for sequence control or error correction. CCITT was sure that data service users would want to interact with the network in the same way they would use a tape drive.

ARPA felt that it did not have time to wait for an international standard, because it needed to immediately interconnect the various networks it was building or designing: these included ARPANET, a shared channel satellite network (SATNET),[40] and several networks of mobile packet radios.[17,41] By late 1974 ARPA was funding multiple efforts to implement the ideas from the Cerf/Kahn paper. BBN first implemented these ideas in a 1975 experiment[42] and first demonstrated them with all three ARPA networks in late 1977.[43] After a period of TCP experimentation, the Internet researchers realized that creating a single super-protocol across network boundaries was difficult and limiting; for example, packetized voice didn't need a reliable protocol. They also recognized that the problem was much simpler if it was split into two parts: a simpler Transmission Control Protocol (TCP) that managed communication between endpoints, and a new Internet Protocol (IP) that routed datagrams between different networks.[44] The TCP/IP specification was formally stabilized in 1979.

Meanwhile, in July 1975, representatives from ARPA, Cyclades, and NPL (plus Alex McKenzie of BBN) were working on the problem of establishing an INWG standard that could be implemented by the existing research networks and submitted to the CCITT for use in public data networks. They hammered out a compromise using the best ideas of the various proposals that had been submitted based on datagrams.[45] However, there was no enthusiasm for datagrams in CCITT, and in 1976 CCITT adopted the circuit-oriented X.25 standard for data communications. INWG members were disappointed, but not really surprised, when CCITT rejected their approach. However, the international research community was shocked when ARPA declared that TCP implementation was too far advanced to restart with the INWG proposal. U.S. and European network research diverged as a result.

TCP Research

Although some of the IMP Guys in BBN's Computer Systems Division of were initially rather cool to TCP,[46] the networking people in BBN's Information Sciences Division were TCP enthusiasts from the beginning. The first TCP was implemented by Ray Tomlinson in BCPL[47] for the TENEX operating system.[20,48] While experimenting with this implementation to send files to a printer, Tomlinson found that data from old connections was getting mixed with data from new connections because of overlapping sequence numbers. This discovery led him to develop a theory of managing sequence numbers; in particular, his theory created a set of rules for when a particular sequence number can safely be reused and when its use is forbidden. His paper remains a standard reference today.[49,50]

The initial TENEX TCP implementation was extremely slow — so slow that Bob Kahn expressed concern that TCP would never amount to anything. Bill Plummer of BBN reimplemented TCP in assembly code and put it into the operating system to improve memory performance by swiftly mapping pages. This TCP was used in experiments with several TCP features such as Desynchronize-Resynchronize (DSN-RSN) and Rubber End-of-Lines (used for record demarcation) that ultimately did not become part of the TCP standard.[51]

In 1979, ARPA solicited proposals to replace the aging TENEX operating system with a new research operating system for the ARPA community. ARPA split the work between two teams: the Computer Science Research Group at U.C. Berkeley, which would implement a paged version of UNIX 32/V; and BBN, which was responsible for all the networking code. This version of UNIX and TCP ran on a DEC VAX minicomputer.

The BBN networking implementation was largely done by Rob Gurwitz, with some help from Jack Haverty. Haverty had already done a TCP implementation for UNIX version 6 on a PDP-11. Although they could have started with Haverty's TCP, they decided to start afresh, in large part because Haverty's version tied TCP and IP closely together (a vestige of the original single-protocol standards).

Gurwitz's implementation, the first widely used UNIX implementation, had quite a few interesting features. The implementation required applications to open special UNIX files (for example `/dev/tcp`) to create network connections. To manage variable-sized packets in memory, Gurwitz created a new type of memory buffer called an `mbuf`. And, in an interesting internal feature, the implementation used a state-event matrix of functions: that is, if you received a particular type of packet, and your connection was in a particular state, you indexed a matrix to find a pointer to the appropriate function.[52]

The BBN BSD TCP was the standard TCP for 4BSD and BSD UNIX 4.1. However, in BSD 4.2, the team at Berkeley created their own and very different implementation of TCP/IP (using the now familiar `socket` interface developed by Bill Joy and Sam Leffler of Berkeley along with Gurwitz). BBN promptly revised its TCP implementation to use the socket interface,[53] and for about a year there was a battle to determine whose networking code would take precedence. Although the BBN code won some adherents and was licensed to several computer vendors, the Berkeley code won the battle.

A few effects lingered, however. First, `mbufs` remained the standard way to manage packet memory until the mid-1990s. Second, and somewhat amusingly, the Berkeley team would sometimes justify bugs in their TCP by pointing out that the original BBN code had the same bug. Third, ARPA continued to fund a vestige of the BBN UNIX TCP project into the late 1980s, and during that time BBNers (Karen Lam, Craig Partridge, and David Waitzman) worked with Steve Deering to create the first implementation of IP multicast.

BBN did TCP implementations on other platforms. Charlie Lynn[54] wrote a TCP for the DEC TOPS-20 system. Jack Sax and Winston Edmond wrote an implementation for the Hewlett-Packard HP-3000 (ARPA was concerned that all the TCP implementations were on DEC machines and wanted to show it was not DEC-specific).

Open Systems Interconnection (OSI)

While BBN's attention was focused on the AUTODIN II procurement and ARPA's TCP program, the European computer community was growing increasingly distressed over CCITT's standardization program. Immediately after adopting the X.25 host interface standard, CCITT began an accelerated program of standards development for support of terminals and applications such as electronic mail. Computer manufacturers doing business in Europe, as well as the computer research community, felt the telephone monopolies must be prevented from controlling the form that computer application software would take. They decided to counter the CCITT by creating a data-communication standardization activity within the International Organization for Standardization (ISO), which had already produced many computer standards. The first meeting of this activity, known as Open Systems Interconnection (OSI), took place in early 1978.

The U.S. government was represented in ISO by the National Bureau of Standards (NBS).[55] The DoD urged NBS to ensure that any standards developed in the OSI project provided the same functionality as TCP (and IP), so that eventually this functionality would be provided by computer manufacturers as part of their bundled software rather than needing to be developed specially with DoD funding. In order to achieve this goal, NBS awarded BBN a contract to provide technical assistance both at and between ISO meetings; this assistance took the form of drafting position papers and detailed protocol specifications that reconciled the NBS interests with the requirements of other ISO members. Alex McKenzie led this effort for BBN. One consequence of this work was that, for a time, BBN had a weekly lunch-table meeting where people on the NBS contract practiced their French for use at coffee breaks and meals connected with standards meetings.

Perhaps not surprisingly, the TCP community considered the OSI project a colossal waste of time — and the virtual circuits of X.25 a major technical error. Thus, there were many debates in the halls of BBN among the groups implementing X.25 interfaces for DCA, routers and TCP/IP host software for ARPA, and OSI proposals for NBS. In spite of these internal debates (or perhaps because of them), the group supporting NBS achieved some notable results. Debbie Deutsch, Bob Resnick, and John Vittal developed a considerable portion of "Abstract Syntax Notation 1" (ASN.1), a structural framework used by ISO, CCITT, and the Internet community to describe the content and encoding of application data.[56] Ross Callon almost single-handedly convinced ISO to include a "connectionless" network facility corresponding to IP in the OSI standards and wrote most of the specification. John Burruss and Tom Blumer developed a method of formally describing a protocol state machine in terms easily understood by human readers, yet directly compiled and executed, thus eliminating the ambiguities possible in a natural-language protocol description. Blumer implemented an execution environment to support the protocol state machine, and Burruss wrote the formal description of a protocol providing the functionality of TCP that ISO included in the OSI standards.

17.3 IP, Routers, and Routing Protocols

Central to the concept of IP is the idea of a *router*, a device that takes datagrams from one network and places them on another network. It is called a router because its job is to move datagrams between networks in such a way that the datagrams proceed along the correct routes to their destinations. Equally important is the concept of routing protocols, by which routers learn from each other how to move a datagram from network to network from its source to its destination.

By late 1980 the DoD had adopted TCP/IP and the ARPANET's terminal support (Telnet protocol) and File Transfer Protocol (FTP) as DoD standards. In 1981 planning began for all ARPANET hosts to transition to TCP/IP. The official transition completion date was to be January 1, 1983; in fact the transition was not completed for several more months.

The conversion to TCP/IP was mandated to make it possible to split ARPANET into multiple networks without disrupting host computers' ability to communicate with one another regardless of which network they were assigned to. The networks were to be connected by *mail bridges*. These devices were customized routers that could filter out undesirable traffic — in essence, the first firewalls. The mail bridges were built and operated by BBN, making BBN the first Internet router vendor, and putting BBN in the center of the early development of IP routing protocols.

Routers

Probably BBN's earliest published thinking relating to building routers resulted from work with satellite networks.[17,57] In 1975 Virginia Strazisar joined BBN and was given the task of implementing an IP router (at that time, called a *gateway*) on a PDP-11. This was a BCPL[47] implementation on the ELF operating system and is remembered fondly as being a wonderful prototype: It ran well, albeit slowly. By late 1976 three routers were up and running: one at BBN connecting an ARPANET clone that BBN used as its internal LAN with the ARPANET itself, one at SRI between the ARPA Packet Radio Network and the ARPANET, and one at University College London connecting the Atlantic Satellite Network and ARPANET.[58,59,60]

In 1981, in anticipation of ARPANET's TCP/IP transition, work on routers in BBN was given to a new team, led by Bob Hinden, charged with developing a router system that BBN could operate for DoD. Mike Brescia and Alan Sheltzer reimplemented the router in assembly language under the MOS operating system for both the PDP-11 and the DEC LSI-11 processors.[61] The LSI-11 rapidly became the preferred platform and was widely used into the mid-1980s.[62,63]

Around 1983 the BBN router team began to grapple with the deficiencies of shared-bus, single-processor hardware as a base for router implementation. In a device whose job is largely moving data between external interfaces, the BBN team felt the most efficient architecture would put processing near each interface and allow interfaces to talk to each other directly, rather than having to go through a processor that managed a shared bus.

This thinking was about a decade ahead of its time and market needs. However, BBN was in a position to make the multiprocessor router a reality. Another team in the company was completing an innovative multiprocessor computer called the Butterfly, which interconnected processors and peripherals through a time-slotted banyan switch. So BBN decided to try to build a next-generation router on the Butterfly platform.[64] The team was led by Hinden and included Eric Rosen, Brescia, Sheltzer, and Linda Seamonson.

As a research activity, the Butterfly router was an important innovation. The software, in C, was the first demonstration that a high-performance router could be implemented in a higher-level language. The router team also learned a number of painful lessons, most notably that the Butterfly was rich in processing power but weak in bandwidth between peripherals and that this balance was exactly the reverse of what a router would want. Indeed, performance issues led Rosen and Seamonson to invent an early version of label switching.[65] Although the Butterfly gateway (as the router was called) was the fastest router available, its performance/price ratio was poor.

Unfortunately, the Butterfly gateway became BBN's de facto router product. It was a mistake. The router was expensive; it was slow to reboot,[66] and while it eventually performed well, it was hard to maintain. BBN managed to sell around 50 of these Butterfly gateways, largely to government clients who needed the fastest router possible. But when the Internet was opened to general use and the router market suddenly blossomed, BBN was caught flat-footed. The Butterfly gateway, although a more mature product, was simply not price competitive.

Despite internal resistance (one vice president asked why he would want to build a \$20,000 router when he was selling IMPs for more than \$80,000), a team led by Bob Hinden and Steve Blumenthal did build a price-competitive router called the T/20 that placed the Butterfly gateway code on a single-processor card, with daughter cards for each interface. The T/20 was used extensively to support packet videoconferencing and distributed real-time simulation on the Defense Simulation Internet. It was also widely deployed in the U.S. Army's Mobile Subscriber Equipment network. However, by the time the T/20 router reached the commercial market, Cisco Systems had already won the race for market share.

Although the Butterfly gateway swiftly faded, its influence lingered. A team that included part of the Butterfly team[67] designed and built an early high-end asynchronous transfer mode (ATM) switch. BBN created a new company, BBN Lightstream Corporation, to manufacture and market the switch. Lightstream was funded by BBN and Ungermann-Bass and was eventually sold to Cisco.

A little later, between 1992 and 1996, a BBN team led by Craig Partridge, Josh Seeger, Walter Milliken, and Phil Carvey (Milliken and Carvey were members of the Butterfly team) designed and built a prototype of the world's first 50-gigabit-per-second router. The router design reflected BBN's painful experience with the Butterfly gateway. It included a switch designed specifically to move IP datagrams from arbitrary input interfaces to arbitrary output interfaces. (One of the lessons of the Butterfly experience was that IP traffic tended to include bursts of datagrams to a single destination, which could overload switches that assumed balanced traffic.) Variants of this router architecture became standard in the router industry, and BBN's paper on the router[68] became required reading at many corporations. BBN remains a center of expertise in the design and implementation of high-end router and routerlike devices (for example, encryptors) today.

Routing Protocols

The government's 1968 ARPANET procurement document[6] asked the contractor to design a routing algorithm for the ARPANET and suggested an example algorithm based on complete knowledge of the network configuration at a central control facility and updates from the central facility to the individual packet switches.

BBN viewed central control as inconsistent with the ARPANET robustness goals and instead designed and implemented a dynamic system that set the stage for the worldwide distributed routing system of today's Internet. Bob Kahn suggested the structure for distributed routing,[69] and Will Crowther devised and implemented a detailed set of algorithms that:[9,70,71,72]

- adapted to changing installations of switching nodes and internode communication links with minimal configuration information in each node and no centralized control;
- discovered and adapted to temporary node and link ups and downs; and
- routed data traffic along the path of least delay.

The implementation included link alive/dead logic, internode packet retransmission logic, and a distributed, asynchronous, adaptive routing calculation. These features were a major break with the more or less fixed routing under central control and inadequate internode data acknowledgment schemes that were typical up to 1969. The implementation included the discovery of the distributed asynchronous real-time algorithm now widely known as ARPANET distance vector routing.[73]

This initial routing could not adapt accurately enough or fast enough as ARPANET (and later the Internet) grew in complexity and size. Nonetheless, it did provide an initial dynamic, distributed implementation that supported the quasi-operational ARPANET in its early years and provided a test bed for developing improved algorithms.[74]

From 1973 to 1975, John McQuillan tuned the initial ARPANET routing algorithm and implementation and began planning an improved implementation.[75] From 1976 to 1979, led first by McQuillan and later by Eric Rosen, a small team designed, experimented with, and finally implemented operationally a new ARPANET routing algorithm[76] now known widely as ARPANET link state routing or shortest path first (SPF).[77] The essence of this implementation[78] was to build a routing database of topology and traffic for the whole net, and to build a complete routing tree at every node. Much work and careful thinking went into ways to make the distributed routing databases accurate and coherent, including development of an improved means for measuring network delay, and flooding to disseminate the information reliably and efficiently. This implementation included a real-time distributed implementation of Dijkstra's algorithm.[79]

When it came time to implement IP routers, BBN built on its prior work. The first routers used a distance vector protocol called the Gateway-to-Gateway Protocol[80] patterned after Crowther's original ARPANET routing protocol. Later, as the Internet grew, GGP had the same scaling problems as its predecessor, so the Butterfly gateways implemented an SPF protocol patterned after McQuillan's.

However, the BBN router team also realized that one routing protocol was not enough. There needed to be a hierarchy of routing protocols — in particular, there needed to be some way to put boundaries between different pieces of the Internet so that errors and routing problems in one part of the network didn't spill over (or, at least, spilled less) into other parts of the network. To share routing information across these operational boundaries required a new type of routing architecture and protocol. Eric Rosen developed an architecture dividing the Internet into a set of autonomous systems, each of which was a relatively homogeneous set of routers and routing protocols, and the Exterior Gateway Protocol (EGP) to communicate basic routing information between the autonomous systems.[81]

The work of Crowther, McQuillan, and Rosen still substantially defines how we do routing today. The major routing protocols (Routing Information Protocol, Interior Gateway Routing Protocol, Open Shortest Path First, Intermediate System-to-Intermediate System, and Border Gateway Protocol) are all derivatives of the original ARPANET protocols and EGP.

17.4 Satellite and radio

Early in the development of the ARPANET, point-to-point links on geosynchronous orbiting satellites were used as inter-IMP circuits to reach overseas locations such as Hawaii and Europe. These links were treated like ordinary circuits in the network, except that they had long delays, typically 250 milliseconds each way, and thus required extra packet buffering and caused problems for the original ARPANET routing algorithm. However, ARPA soon began to investigate ways to use wireless technologies, both satellite and terrestrial, as the basis for building networks.

Early Packet Satellite Network R&D and SATNET

At about the same time that the ARPANET was being built, Norm Abramson and his colleagues at the University of Hawaii developed the "Aloha system," a shared-channel ground-based radio system to provide terminal access to a time-shared computer.[82] Shortly, Larry Roberts at ARPA began talking about research and development using shared satellite channels in the same way. Roberts, people at BBN, and others began writing working notes on such use of satellite channels. The new methods under consideration took advantage of the broadcast nature of the satellite channel to merge uplink traffic from many nodes, use the satellite channel to achieve statistical multiplexing, and deliver the combined aggregate data stream to all sites simultaneously. Selective addressing of the data packets allowed messages to be multicast to a subset of the sites or broadcast to all sites. Two of the working notes written during the early days of these discussions and thinking were breakthroughs, influencing much later work (and undoubtedly influencing the development of Ethernet); they were later republished in the ACM SIGCOMM *Computer Communications Review*.[83] As talk within the working group turned to reserving slots in a shared satellite channel, BBN people published a paper using the name Reservation Aloha (R-Aloha, often pronounced "our Aloha").[84] This annoyed the rest of the participants, who thought BBN's proposed system should have been called something more modest such as BBN Aloha rather than implicitly claiming the idea of reservations, which others also were proposing in their own algorithms.

In the early days of these technical discussions, BBN played a key role. In particular, Dave Walden managed BBN's activities in the area, with the actual work being done by Randy Rettberg, Will Crowther, Steve Blumenthal, and eventually Dick Binder (who joined BBN's team after getting his master's degree helping with the Aloha system research and development at the University of Hawaii).[85] However, after Larry Roberts left ARPA, Bob Kahn was dissatisfied with the concentration of network design at BBN and took active steps to make the program multi-institutional. He involved Irwin Jacobs and Andy Viterbi of Linkabit as well as Estil Hoversten (who later moved to Linkabit), giving Jacobs the lead role in the program. Others who were involved included Dave Mills, Kim Kaiser, and Stan Rothschild from Comsat Labs.

Eventually a scheme called Priority Oriented Demand Assignment (PODA) was developed.[86] PODA divided time on the satellite channel into frames and used one part of the frame to transmit data and another part of the frame to send reservation requests. Short reservation-request messages were sent in slots in the reservation part of the frame. These requests were received by all nodes listening on the satellite channel. All nodes performed a distributed scheduling algorithm and assigned a portion of the data part of a subsequent frame to the site that had been given the reservation. At the reserved time, that site would send its data and the other sites would be silent. On the downlink from the satellite, the data would be delivered to all sites on the channel and if the data was addressed to a host computer attached to that node, the data would be delivered; otherwise it would be discarded. Two variations of PODA were developed: Fixed PODA (FPODA) had a reservation slot for each site on the satellite channel, and Contention PODA (CPODA) had a smaller number of reservation slots, and the sites used the slotted Aloha approach[83] to contend for reservation slots. CPODA provided more efficient utilization of the satellite channel resources, though it was harder to implement and get to work.

In 1975, ARPA initiated a project to build a working PODA network named The Atlantic Packet Satellite Network, which was later shortened to SATNET. The purpose of SATNET was to extend the Internet to Europe and to support experiments in the use of broadcast satellite channels for packet switching, as well as joint NATO experiments in distributed command and control. BBN was selected to implement the Satellite IMP as a modification to the standard ARPANET IMP. The SATNET Satellite IMP (SIMP), originally implemented on a Honeywell 316 minicomputer, contained more packet buffer memory to account for the long 250-msec transmission delay between the earth and the satellite. It also had a special hardware interface to the satellite channel burst modem/codec that could precisely time radio transmissions and recover satellite channel clock times. The software implemented the PODA channel access and sharing algorithms. Comsat Labs had responsibility for the earth terminal equipment, and Linkabit Corp. had responsibility for the burst modem/codec equipment. In addition to implementing the SATNET SIMP software and special hardware, BBN had overall responsibility for SATNET operations. The network was monitored and controlled by the ARPANET Network Operations Center at BBN, using some of the same tools that were used for the ARPANET, or tools that had been modified for the SATNET application.

The original SATNET used a 64Kb/s channel on Intelsat's Atlantic Ocean Region and included three main sites located at Intelsat country earth terminals in the United States, Sweden, and the United Kingdom. SATNET was operated as a separate network with early IP routers connecting to ARPANET in the United States and local in-country networks at research institutions in Europe. In the United Kingdom, local area networks at the Computer Science Department of the University College London and the Royal Signals and Radar Establishment connected via Internet gateways to the SATNET SIMP at Goonhilly Downs. In Norway, the Norwegian Defense Research Establishment and the Norwegian Telecommunications Research Establishment connected to the SIMP at Tanum, Sweden. The point-to-point transatlantic link of the ARPANET was disconnected, and SATNET provided the primary network connection of the emerging Internet to Europe. In the early 1980s two new SIMP sites were established at Raisting, Germany, and Fucino, Italy, connecting the German Aerospace Research Agency (DFVLR) located outside Munich and the Italian Computer Research Center at CNUCE in Pisa.

In the early 1980s the SIMP code was ported to the new BBN C/30 packet switch and a special satellite channel I/O card was implemented for the C/30. To increase throughput, and to take advantage of the increased processing power of the BBN C/30, the SIMP code was rewritten to use two parallel 64Kb/s Intelsat satellite channels. As fiber optics became more prevalent and cheaper at the end of the 1980s, SATNET was retired and replaced by transatlantic point-to-point undersea fiber connections.

Wideband Packet Satellite Network and Packet Voice

In the mid-1970s, Bob Kahn at ARPA began to think about how packet switching could be extended to other types of communications, such as voice communications. He commissioned a study by Howard Frank and Israel Gitman of Network Analysis Corp. to examine the economic costs and relative efficiencies of carrying voice by circuit-switching and packet-switching techniques. This study concluded that packet switching had the potential to be substantially more efficient. However, two major problems had to be solved to make packet voice a reality.

First, it was necessary to find a way to digitize voice into packets. John Makhoul and the BBN Speech Group[87] participated in ARPA-sponsored research in voice compression techniques that led to the development of linear predictive coding (LPC). LPC has become one of the standard ways for voice to be compressed and transmitted as packets. Based on perceptual studies, it was determined that chopping digitized voice up into packets every 20 msec would be effective. Pulse-code-modulated speech, sampled at a rate of 8,000 samples/sec at 8 bits resolution (a 64Kb/s data rate with no compression), resulted in packets that had 160 bytes of data. The actual packets were a bit longer because of the addition of a few bytes of routing header information. (As of 2003, typical LPC voice coder/decoders [vocoders] ran at 8–10Kb/s.)

Second, as early experiments in sending voice packets over the ARPANET revealed, voice transmission required a pure datagram service. The ARPANET, with its “Ready for Next Message” (RFNM) signaling technique, was all about reliable end-to-end delivery of information. New packets could not be sent into the network until outstanding packets that had already been sent were acknowledged as having successfully reached their destination host computer by a RFNM message. This did not work for voice. Voice packets needed to be sent without waiting for an end-to-end acknowledgment. If a small number of packets were lost along the way or arrived out of order and were discarded by the receiver, the human ear and brain could interpolate through the missing audio information and understand what was being said. If a large number of packets were lost, listening to the voice could become tiresome and the speech could become unintelligible. When users were trying to have a two-way conversation, they were also sensitive to the end-to-end delay of the voice packets. Experiments showed that when end-to-end delay rose above 200 msec, users felt that they were participating in a “half-duplex” conversation in which people had to be very deliberate in turn-taking and not interrupt each other. It became obvious to the early packet voice researchers that the underlying packet-switched network would have to deliver a certain level of service — low packet loss, low end-to-end delay, and low interpacket jitter (the time-difference of arrival of voice packets at the receiver — for users to feel that packet voice was an acceptable substitute for the circuit-switched voice that they were used to with the telephone system.

ARPA IPTO realized that it would be necessary to create a packet-switched network that could deliver the right level of quality of service (QoS) to carry voice along with data traffic. Even with compression algorithms such as LPC, to get a meaningfully large number of packet voice calls to intermix with packet data would require a much higher-bandwidth network. High-speed terrestrial circuits such as T1 phone lines that ran at 1.5Mbps were still very expensive and difficult to obtain in the late 1970s. Dick Binder of BBN, Estil Hoversten and Irwin Jacobs of Linkabit Corp., and Bob Kahn and Vint Cerf from ARPA conceived the Wideband Packet Satellite Network (Wideband Net) as an appropriate vehicle to deliver the necessary QoS. This network was built around a 3Mbps broadcast satellite channel that connected multiple sites in the United States and supported broadcast and multicast delivery for voice conferencing. The Wideband Net added packet speech to the packet satellite data-networking experiments by creating a stream service for packet voice traffic. The stream service permitted sites to reserve periodic time slots in each frame on the satellite channel to carry the voice packets. The rest of the frame could be used for more bursty data traffic. The first four sites were the MIT Lincoln Laboratory in Lexington, Massachusetts; the Defense Communications Engineering Center (DCEC) in Reston, Virginia; USC ISI in Marina del Rey, California; and the Stanford Research Institute (SRI) in Menlo Park, California.

Binder led the BBN team that built the original Wideband Net packet switch; Gil Falk took over the project when Binder left BBN. The switch was built on the BBN Pluribus multiprocessor and was called the Pluribus Satellite IMP or PSAT for short.[88] The Pluribus was chosen because with about a half-dozen Lockheed SUE minicomputer processors and a high-speed satellite interface, it had the processing power to run the PODA algorithms and keep up with the 3Mb/s channel. The Pluribus presented a very difficult programming environment. During the project's early stages, many people — including John Robinson, Tony Lake, Jane Barnett, Dick Koolish, Steve Groff, Walter Milliken, Marian Nodine, and Steve Blumenthal — worked on the implementation. Burnout on the programming team was a problem. The PSAT software was buggy and could not be made to run reliably for any lengthy period of time. The hardware's several wire-wrapped boards also had some long-term stability problems and faults that were difficult to isolate to specific hardware or software causes. Blumenthal took on an operations role and began to figure out how to make the PSAT and the entire system more robust overall, and eventually became the Wideband Net project manager.[89]

In the late 1970s, as a part of the Wideband Net effort, BBN began to develop a high-performance packet voice multiplexer called the Voice Funnel. The Voice Funnel would be based on the Butterfly multiprocessor conceived by Will Crowther, Randy Rettberg, Mike Kraley, John Goodhue, Phil Carvey, and Bill Mann at BBN.[90] The Butterfly would have processor nodes with memory connected by a switch network that was patterned after the butterfly network used for the Fast Fourier Transform algorithm. The machine was based on the notion of shared memory (all memory was accessible to each processor) and was highly scalable. Butterfly machines with as many as 256 processor nodes were eventually built. I/O modules were attached to specific processors. The initial Butterfly used the Motorola 68000 microprocessor; more importantly, it had a real operating system, called Chrysalis, running on every node and a process-oriented programming model with a process scheduler. The Butterfly supported the C programming language. The Butterfly had the performance to handle the Wideband Net packet-switching task, and Randy Rettberg and Steve Blumenthal were able to convince Bob Kahn and Barry Leiner at ARPA to let BBN port the PSAT to the Butterfly to create a BSAT. Winston Edmond had joined the Wideband Net, and he headed up the architectural design of the BSAT. Edmond created an elegant design that took advantage of situations where the PODA satellite channel-scheduling processes could be parallelized for increased performance. The BSAT was begun in 1982, and Milliken and Nodine worked with Edmond to complete it by 1984.[91]

The Butterfly proved to be much easier to program than the Pluribus, and the BSAT was a big success. Steve Blumenthal and the BBN team turned their attention to how to get the Wideband Net to work on an end-to-end basis. Working cooperatively with ARPA, Linkabit and Western Union were able to get the Wideband Net to meet the ARPA's functional and performance goals and to operate very reliably. Users were now reliably supported in their packet voice and video conferencing and high-speed networking research. An additional six sites were added to the network. User organizations such as MIT Lincoln Laboratory, MIT Laboratory for Computer Science (LCS), USC ISI, and SRI began to use the Wideband Net for multisite packet voice and video conferencing. The Diamond Multimedia Mail group under Harry Forsdick at BBN began to experiment with shared-workspace multimedia conferencing across the Wideband Net.[92] Using high-performance Butterfly gateways, the Wideband Net was extended to interface to 10Mb/s Ethernet local area networks (LANs) and nearby sites such as the MIT and Stanford University campuses using high-speed T1 (1.5Mb/s) phone circuits and microwave links. The MIT LCS group under Dave Clark used the Wideband Net to develop a new high-performance, high-volume file-transfer protocol called NETBLT. It transferred large blocks of data over high-speed high-latency paths by sending the whole data set and then resending missing or corrupted pieces at the end. The Wideband Net also used to provide real-time networking between SIMNET sites.[93]

By the mid-1980s, the Wideband Net's stream service support of resource allocation and quality of service (QoS) within the network led to the development of similar capabilities at the Internet level. Claudio Topolcic and Lou Berger of BBN led the development of the ST-2 protocol to support real-time packet voice and video communication over the Internet. These protocols were developed under the auspices of the newly-formed Internet standards body called the Internet Engineering Task Force (IETF). The ST-2 protocol, a connection-oriented protocol that maintained state within the network, and other connectionless schemes such as the Resource Reservation Protocol (RSVP), Real Time Protocol (RTP), and Differentiated Services (DiffServ) also were developed by the IETF as alternatives; BBN people were contributors to all of these efforts. These protocols form the basis of Voice over IP (VoIP) services today.

As T1 phone line service became cheaper and more widely available in the late 1980s, high-speed terrestrial networks could be built that did not have the 250-msec satellite channel latency. Winston Edmond adapted the BSAT software to work over a shared cross-country bus made up of parallel T1 circuits. This network became known as the Terrestrial Wideband Network (TWBNET) and the BSATs became Wideband Packet Switches (WPSs). The TWBNET supported a real-time stream service along with a bursty datagram service. In the early 1990s BBN extended this network globally, from Germany to South Korea, as the Defense Simulation Internet, which supported real-time SIMNET and other war gaming exercises.

Gigabit Satellite Networking Using NASA's ACTS

In 1992 ARPA and NASA selected a team led by Marcos Bergamo to design, develop, deploy, and operate the world's first Gigabit Satellite Network (using NASA's Ka-band Advanced Communications Technology Satellite, or ACTS). The goal was to demonstrate the practical feasibility of integrating satellite and terrestrial Internet and Asynchronous Transfer Mode (ATM) switching services to support distributed supercomputing, remote visualization, and telemedicine applications. The initial architecture, performance requirements, challenges, and development recommendations for the network were first defined in a study BBN prepared for ARPA during the early 1990s.[94] A network of five transportable Ka-band (20/30 GHz) earth stations, built around the gigahertz-wide multiple-beam-hopping and on-board transponder-switching capabilities of ACTS, was completed on a tight two-year schedule.[95]

Bergamo and his team decided on an approach that integrated SS-TDMA (Satellite-Switched Time Division Multiple Access) and OC-3/OC-12 SONET add/drop multiplexing at the earth stations.[96] This approach involved many challenges, including the need to:

- develop a 120-watt traveling wave tube amplifier and a 3.4-meter Ka-band antenna;
- design a near-gigabit-rate modem capable of operating over gigahertz-wide transponders with then-unfamiliar noise-saturated characteristics;
- design and develop a digital terminal capable of multiplexing OC-3 (155 Mbps) and concatenated OC-12 (622 Mbps) SONET data into satellite-switched TDMA bursts;
- invent a way to synchronize and distribute SONET data gathered from diverse locations (geographically distributed SONET islands in the continental United States and Hawaii); and
- engineer the critical issue of initially acquiring, and then maintaining, earth station synchronization with the microwave and beam-switching on board the ACTS satellite.

The network was deployed to five U.S. sites in 1994–95 and operated until April 2000, when the ACTS satellite was decommissioned. During its lifetime the ACTS Gigabit Satellite Network was used for experiments that included a distributed supercomputing Lake Erie weather simulation, remote operation and visualization of the Keck telescope in Hawaii by astronomers at NASA Goddard, and multiple integrated ground-satellite Internet/ATM test beds. In 1997 key BBN developers of the Gigabit Satellite Network were inducted as "satellite innovators" to the U.S. Space Technology Hall of Fame, and for his work Bergamo was personally recognized with the 2005 IEEE Judith Resnik Award.

Packet Radio, Wireless and Tactical Military Networks

BBN has been a major contributor to terrestrial wireless networking, particularly in mobile ad hoc networks — also called packet radio networks or multihop wireless networks. An ad hoc network is a (possibly mobile) collection of wireless communication devices that communicate without fixed infrastructure and have no predetermined organization of available links. The lack of fixed infrastructure, rapid changes in connectivity and link characteristics, and the need to self-organize pose challenges that make ad hoc networking significantly more difficult than cellular networking.[97]

In 1973 ARPA started a "theoretical and experimental" packet radio program. The initial objective was to develop a geographically distributed network consisting of an array of packet radios managed by one or more minicomputer-based stations, and to experimentally evaluate the performance of the system. The first packet radios were delivered to the San Francisco Bay area in mid-1975 for initial testing, and a quasi-operational network capability was established for the first time in September 1976.[98] The project was a multi-institution project led originally by Vint Cerf at ARPA and later by Barry Leiner. Rockwell International/Collins developed and manufactured the packet radios and contributed some ideas to the overall program. SRI did the initial system design and ran the program and the testing. Jerry Burchfiel and his team[99] in BBN's Information Sciences Division developed the gateway to connect the packet radio network to the ARPANET and SATNET (developed by Virginia Strazisar and previously mentioned on page 426) and developed the centralized (later distributed) routing and management stations.[41] Jil Westcott remembers that Rockwell's radios were so expensive that ARPA recompeted this part of the program and Hazeltine won; however, Hazeltine couldn't meet their cost goals and the radios didn't work well.

The ARPANET team in the Computer Systems Division was jealous that this project in "their" domain of packet switching (where they were already doing the ARPANET and satellite network work) had gone to the Information Sciences Division. Many of them felt that Bob Kahn at ARPA was being vindictive to his old ARPANET colleagues, especially Frank Heart, for battles he didn't win during the IMP implementation. Jil Westcott points out,[100]

> Communication between the divisions was poor, and technical work done by [the packet radio team] was not looked at closely by others [who had already been looking at similar issues]. . . . The large-scale routing algorithms [were] the primary case in point. We got funding to explore this topic. . . . ARPA liked having two different parts of BBN competing for networking ideas and supported our independence. . . . At one point we were given an ARPA-sponsored project on network management that no one had done much with, and we turned it into something quite valuable for Packet Radio. We aimed the program at the soldiers in the field and worked with [BBN's] human factors people to make it easy to use. [It w]as quite graphical and did well in field tests at Fort Bragg. This system was taken up by BBN Planet and used for many years to manage their networks. [It w]as also bought by a French company, who productized it and made a bundle. However, the product part of BBN [BBN Communications Corporation] could not be interested in using this software.[101] We later tied [our] network management [system] into [the work of BBN's main] network analysis group by using their underlying tools to create an excellent analysis product for looking at network behavior. [This ultimately] failed long-term as we relied upon Symbolics to provide the LISP machine on a board which could be plugged into a Sun box for a low-cost delivery solution. Symbolics went out of business and never delivered the production board, and the code was too hard to rewrite outside of LISP.

In the mid-1980s BBN played a key role in the next phase of the ARPA packet radio thrust, the development of the Survivable Adaptive Radio Networks (SURAN) program[102] — the first comprehensive prototype system for battlefield networking of elements in an infrastructureless, hostile environment.

Under subcontract to GTE Government Systems, BBN designed and built a packet-switched overlay network for the U.S. Army's Mobile Subscriber Equipment (MSE) tactical radio communications system using ruggedized versions of the BBN C/3 packet switch and the T/20 IP router. This was a huge contract for BBN during the late 1980s and early 1990s. Thousands of C/3s and hundreds of T/20 IP routers were deployed, using Army tactical radio links to provide field data services. The MSE contract gave BBN the opportunity to further refine its program-management skills for large government systems and also gave it credibility in the tactical communications arena.

In the early 1990s BBN played a crucial role in two programs. One was the U.S. Army's Near Term Digital Radio (NTDR), for which BBN developed scalable, adaptive networking.[103] The second was the ARPA Global Mobile Information Systems (GloMo), as part of which BBN completed two large projects — the Mobile Multimedia Wireless Network (MMWN)[104,105] and the Density- and Asymmetry-adaptive Wireless Network (DAWN).

The NTDR (also called ITT's "Mercury" radio) represented a real first for the ad hoc networking community. It was the first complete IP-based mobile ad hoc network designed, built and actually fielded. The radios interacted with off-the-shelf IP routers to provide interoperability with all IP routing protocols running on any attached subnets. The NTDR network self-forms, self-heals, and continually self-organizes as the vehicles it is installed in move at up to 65 mph. The radio system also includes a sophisticated networking management terminal that allows the visualization of the ad hoc network and node mobility on an Army-certified terminal. It underwent extensive field tests for many years as part of the Army's "First Digitized Division" (the Fourth Infantry Division at Fort Hood). In December 2002, the NTDR was certified to be ready for the field and participated in Operation Iraqi Freedom. An update to the NTDR called the HCDR (High Capacity Data Radio) is currently being installed and deployed by the U.K. Army as part of their Bowman program.

The NTDR was the prototype program of what was to become an Army-wide program called the "Future Digital Radio." Unfortunately, this changed with new congressional requirements for cross-service interoperability with legacy radios and software radio upgradability. Out of this decree the Joint Tactical Radio System (JTRS) program was developed. The first few years of the JTRS program included the development of a government- and industry-wide software architecture. Once this architecture was developed, there were a number of "experimental" programs to test out the concepts. BBN teamed with BAE Systems to develop the JTRS-Step 2C radio (BAE) and networking protocols (BBN). The hierarchical clustering ad hoc protocols for this 2C radio were similar to those of the NTDR, but the fundamental difference was that this was the *first* ad hoc network to be compliant with the new JTRS Software Communications Architecture (now a requirement for all DoD radios). This early leadership in the JTRS program allowed BBN to join the Boeing-led team for the JTRS-Cluster I program (later renamed JTRS Ground Mobile Radio or JTRS GMR). The GMR program is the first to fully outfit all of the ground vehicles and helicopters of the Army with a new ad hoc networking radio. As opposed to the experimental 2C program, the GMR program is an actual deployment program. BBN is again developing the multihop protocols, also called the Wideband Networking Waveform (WNW), which require scalability to 1,600 nodes in a single area. The first JTRS GMR units were tested in early 2005. User tests were continuing in 2010, anticipating eventual operational test and evaluation in 2012 before full production.[106] Follow-on systems such as as the JTRS AMF (Airborne, Maritime, and Fixed) for the Navy and the Air Force are also expected to use BBN's WNW protocols. In summary, most operational ad hoc networks in the military today uses BBN software, and it appears that as more ad hoc networks are deployed over the next few years, BBN software will be used in many of those systems.

BBN's emergence as the leader in ad hoc networking protocols is due to the ground-breaking work done through the research programs from the 1970s to the 1990s. Today BBN is actively involved in several DoD programs for the next generation of battlefield networking. For example, as part of the ARPA Future Combat Systems (FCS) communications program, BBN developed the networking for an ad hoc network with directional antennas and demonstrated a prototype network with 20 nodes. This was the first prototype of a directional-antenna-based ad hoc network of any size and has given BBN unique expertise in the area of directional-antenna-based battlefield networking. This network included ground vehicles, a helicopter, and heterogeneous customer radio systems on each rapidly moving node. It is not an exaggeration to say that BBN today has become a primary go-to organization for military wireless networking needs.

BBN also has been a thought leader in ad hoc networking research, often opening up new research avenues that the community is now hotly pursuing. In particular, BBN has achieved recognition as the leader in the use of directional antennas for ad hoc networking,[107,108] the development of energy-conserving cross-layer protocols,[109,110] the concept of topology control,[111,112,113] and scalable routing.[114]

Directional antennas allow longer ranges and better spatial reuse of the spectrum, and promise much higher performance, than the traditional omnidirectional antennas. Ram Ramanathan's 2001 ACM/MobiHoc paper laid out the initial concepts on the magnitude of the potential of directional antennas and their exploitation in ad hoc networks. In an award-winning paper,[115] Jason Redi and Ramanathan described the first-ever real-life prototype for directional-antenna-based ad hoc networks. In 2004 Redi and others designed an ad hoc network system that was able to function with one percent of the energy usage of then existing systems, while not sacrificing other performance aspects such as throughput. One of the keys to this reduction is one of the first protocols to allow for distributed slot synchronization without the use of external signals such as GPS.[116] In 2006 BBN developed a unique partnership with Army Research Labs to host and field these low-energy protocols on Army sensor radios. Topology control is the idea of adjusting node parameters in a dynamic and distributed fashion so that the ad hoc network maintains the topology that is best suited for a given objective (for example, a network that both is robust and has a high capacity). In Ramanathan and Regina Hain's seminal 2000 Infocom paper and follow-up Ramanathan and Hain WCNC and Ramanathan 2001 Milcom papers, BBN laid the conceptual foundation and described a variety of solutions for topology control. Finally, the hazy-sighted link state protocol[117] developed by BBN is an innovative, more scalable variant of the routing protocol for ARPANET and is being used in ongoing military programs.[118]

17.5 Secure networks

Because of BBN's position as developer and operator of the ARPA packet-switching network, it was natural for the company to become involved in research and development for how to provide security for data flowing over a packet-switching network, starting in the early 1970s. A little later in the 1970s, Steve Kent finished his PhD at MIT and joined BBN, where he is still a leading contributor to BBN security research.

PLI

BBN's first packet-switching network security approach was the Private Line Interface (PLI). Steve Kent has said,[119]

> The first packet encryptor was the PLI developed in the early 1970s by BBN under ARPA funding. It was approved by NSA for limited deployment on the ARPANET, to protect classified data being sent by DoD folks, starting in 1975 (a somewhat more sophisticated version was approved for use in 1976). Due to the restrictions imposed by use of government COMSEC equipment (KG-34), this was a manually keyed system.

Bob Bressler remembers,[120]

> The fundamental premise [of the PLI] was that the message could be broken into two parts — the header and the data. Transmitting the header in the clear was necessary to enable the network to correctly route the packets, but the data was encrypted. The packets were padded out to maximum length before encryption to prevent the length of the message being used as a signaling mechanism. This scheme was designed for point-to-point use [across the network], so the encryption schemes [at

> each end of the "circuit"] could be synchronized in an off-line, out-of-band manner. Special hardware was used to connect the "red" side of the PLI to the encryption box to the "black" side. The special hardware split the header from the data and bypassed the encryption for the header. The "bypass" was intentionally bandwidth limited to prevent that path being used to inadvertently pass data.

PLIs were used in the COINS network.[121]

BCR

BBN's next network security R&D effort was the BCR (black-crypto-red). Of the BCR, Steve Kent has said,[119]

> In the 1975–1980 timeframe, BBN and the Collins Radio division of Rockwell developed and did limited deployment of the BCR, also under ARPA funding, as an experimental network encryption device. The BCR worked in the TCP/IP protocol environment, used the first NBS-certified DES chips, and had automated, KDC-based key management and access control (the same model later adopted by Kerberos and Blacker). The BCR underwent substantial performance testing in 1980–81, before being retired.[122]

Quantum

Starting with development of the PLI and BCR, BBN has now spent more than 30 years doing research and development relating to network security.[123] Here we will give one modern example.

In the autumn of 2001, DARPA commissioned BBN to build the DARPA Quantum Network, the world's first network protected by quantum cryptography.[124] Unlike existing cryptographic techniques, quantum cryptography bases its ultrahigh security on the laws of quantum physics, and in particular on the preparation, modulation, and detection of single photons or pairs of entangled photons. The first link in this network became operational on December 5, 2002, and has remained in continuous service ever since. Together with academic partners Harvard University and Boston University, BBN is now building a larger suite of quantum cryptographic gateways based on a variety of quantum physical phenomena including the Heisenberg uncertainty principle, the "no-cloning theorem," and entanglement. The network was deployed through dark fiber in the metro Boston area starting in the fall of 2003, and with fielding of specialized optical switches and eavesdropping-aware routing protocols in June 2004. As of June 2006 BBN had a total of 10 nodes operating across the city, performing quantum cryptography both through telecom fiber and through the atmosphere. BBN is also leading a parallel effort to test this network against sophisticated eavesdropping attacks.[125]

17.6 Network operations

Typically, once BBN built an innovative network such as ARPANET or the WIDEBAND network, BBN was asked to operate the network. BBN rapidly developed a strong competence in network operations. It had a 24/7 network operations center (NOC) for the ARPANET, supplemented by on-call technical staff. The ARPANET NOC was a major resource and was often called upon to help operate other networks such as SATNET and the WIDEBAND network.

Although the operations staff was professional, the community was small enough to tolerate (indeed, encourage) a bit of flair and friendly one-upmanship. Mike Brescia always seemed to know what was happening in any router on the Internet.[126] Dennis

Rockwell specialized in debugging knotty network problems and always seemed to know (by heart) the phone number of the relevant techie whose system was causing the problem. The result was a fun-loving yet very earnest group of people who ran many of the world's biggest data networks into the 21st century.

CSNET

By the late 1970s, computer scientists at U.S. universities had realized that computer science departments on the ARPANET were able to collaborate and share research ideas at an entirely different speed than departments not on the net and that nonnetworked departments were in danger of being left behind. CSNET was the proposed solution to this communications "divide." CSNET, created in 1981 by the U.S. National Science Foundation (in cooperation with ARPA), provided e-mail and TCP/IP access to ARPANET to computer science research institutions that did not qualify to be attached to the ARPANET. Building on its success operating the ARPANET, in 1983 BBN won the contract to operate CSNET, taking over from a team of universities that had gotten the network started.[127]

CSNET was the first real Internet Service Provider. Its contract with NSF required CSNET to be self-sufficient after a five-year startup period, so from its inception CSNET was run as a (not-for-profit) business and charged fees from the sites it connected.

The CSNET team at BBN was led by Dick Edmiston. Laura Breeden headed marketing, user services, and accounting, and Dan Long headed technical services. Breeden's team (which for much of the project was just three people) put out a quarterly newsletter, handled dozens of messages a day from academic users trying to figure out how to send e-mail to their colleagues,[128] and worked hard to recruit new sites.

Long's team ran the network, providing 24/7 coverage. Beyond maintaining equipment and dealing with network outages and balky e-mail queues, this team also wrote or maintained much of the CSNET subscribers' networking software. The team also did work for the Internet community as a whole. For several years Long maintained the MMDF-II (Multi-channel Memo Distribution Facility) e-mail software. Craig Partridge figured out how to route e-mail using domain names.[129] Leo Lanzillo wrote the first dial-up IP system.[130]

CSNET was a tremendous success. By 1985 its ARPANET IMP was one of the three busiest in the network. Thanks to tight fiscal management by Edmiston and Breeden, CSNET was self-sustaining by 1986. Soon thereafter, CSNET hired John Rugo as a full-time marketing director, and he began signing up new CSNET members at a prodigious rate.[131]

CSNET's success encouraged NSF to create NSFNET in 1987. NSFNET was a high-speed backbone designed to interconnect regional networks established through start-up funding from NSF. As part of the NSFNET program, BBN was tasked to create the first interNIC (Internet Network Information Center), called the NSF Network Service Center (NNSC). For the first few years of the NSF network program, the NNSC staff handled the cross-network coordination of information for users and regional network operators.

However, the rise of NSF-funded regional networks led to the end of CSNET. In particular, CSNET's e-mail-only customer base swiftly evaporated (why buy e-mail through CSNET when you could get full IP access from a local ISP?). A few years later, CSNET merged with BITNET, and not long after they faded away.

BBN as an ISP

As CSNET began to falter, the CSNET team moved on to a new activity: operating the NSF-funded New England Academic and Research Network (NEARNET), a regional network, on behalf of a university consortium consisting of Boston University, Harvard, and MIT. The NEARNET team was led by John Rugo (who reported to Edmiston) and rapidly repeated CSNET's success. Unlike many NSF regional networks, NEARNET rapidly become profitable.[132] In the early 1990s, the acceptable use policy (AUP)

Figure 17.2. NEARNET Network Operations Center at 10 Moulton Street, Cambridge; left to right are Hannah ?last-name?, Andy Roche, Chuck Miksis, Tom Allen, Cindy Soulia, and Brian Brock. (Photo courtesy of BBN.)

of the NSFNET was changed to allow commercial use of this portion of the Internet. The universities that had been running the NSFNET regional networks began to get out of the network operations business, and BBN acquired other regional networks in northern California and the southeastern United States. In 1995 BBN began to build out a national Internet backbone and, under the name BBN Planet, became the country's second largest ISP.

Operationally, BBN Planet continued in the BBN tradition.[133] Independent measurement services routinely rated its backbone performance as the best (lowest latency and packet loss).[134] Planet rolled out early (and popular) managed firewall and web hosting services. It also managed a considerable portion of AOL's dial-in links for many years.

Internet usage went through phenomenal growth in the late 1990s; for example, BBN Planet was more than doubling in size and quadrupling in traffic annually.[135] As a result, ISPs had to invest heavily and swiftly in new infrastructure simply to avoid losing market share. The need to access large amounts of capital and to keep up with other ISPs caused BBN to be sold to GTE in 1997. GTE married the BBN Planet ISP with a large project to build a global fiber network to create GTE Internetworking. In 2000, when GTE merged with Bell Atlantic to create Verizon, the Internet and fiber network business was not permitted to operate in the former Bell Atlantic territory, and was spun out as a new company, Genuity. From 2000 to 2002 Genuity continued to grow, reaching $1.2 billion in revenue. However, the Internet boom slowed, and Genuity struggled to get its costs in line with revenues. It eventually went through a bankruptcy process and was sold to Level 3 in 2003.[136]

17.7 What has BBN's role been?

This chapter is only an interim history. BBN continues to be a vital source of ideas in many areas of networking. Still, we conclude with a brief attempt to place BBN's contributions of the 1970s and 1980s in context.

The key debatable question, is why didn't BBN capitalize more successfully on its networking leadership in the 1970s and 1980s? In the mid-1980s, BBN was the leading manufacturer of data communications switches and routers and the leading Internet ISP. By the early 1990s, BBN was no longer in the switch and router businesses and was fighting for market share as a leading ISP. Why?

We suggest that the primary reason is a mismatch between BBN's core business model and the style of business required to succeed in the router and switch business. BBN's specialty is contract research — creating new technologies never seen (in some cases, never envisioned) before. That's a labor-intensive and intellectually demanding business. Furthermore, most funding sources fund research by paying for researcher's time at an hourly rate. Accordingly, BBN's research core makes money by recruiting extremely talented people and then finding customers with new problems who can keep those people busy.

In contrast, selling routers or switches is a process of creating standardized (or semistandardized) products that can be sold repeatedly — as a commodity. A product sold in volume does not require substantial additional technical effort.

In this light, BBN's experience of the 1980s and early 1990s makes sense. During the 1980s, data networks were custom products, and BBN built a business of customizing its routers and switches to individual customers, then maintaining the customers' networks. These labor-intensive activities fit reasonably well with BBN's focus on keeping employees busy doing work for customers. When data communications became a commodity, BBN's business focus no longer fit the market, except in the ISP business, where the customers were paying for BBN to operate a network (again, a people-intensive business).

Although BBN had indifferent success at capitalizing on its ideas, there's no doubt that BBN succeeded at transferring its key ideas into the marketplace. We've noted repeatedly how BBN's ideas have become centerpieces of today's networking. BBN has also been a remarkable source of networking talent for the field. Although many individuals mentioned in this chapter are still at BBN, others eventually left to work elsewhere in data communications. Indeed, it is hard to find an important data communications company that does not, somewhere among its key staff, have a few ex-BBNers.

Acknowledgments

This chapter has benefited tremendously from the contributions of current and former BBNers: Mike Brescia, Brian Brock, Bob Bressler, Bernie Cosell, Chip Elliott, Rob Gurwitz, Frank Heart, Bob Hinden, Steve Kent, Dan Long, John McQuillan, Bill Plummer, Ram Ramanathan, Eric Rosen, Jason Redi, Peter Sevcik, Ed Starr, Ray Tomlinson, Ginny (Strazisar) Travers, David Waitzman, and Jil Westcott. The authors thank and apologize to anyone whose name we have forgotten to list. We also take responsibility for any errors of fact.

The list of authors for this paper is in alphabetical order. The first full draft of this paper was prepared by Dave Walden, with initial input from Craig Partridge. Craig, with help from Steve Blumenthal, augmented and transformed that draft into a paper published in *IEEE Annals of the History of Computing*, vol. 28, no. 1, January–March 2006. Alex Mckenzie started with those materials plus his own memories to produce a new draft for this chapter, which was then edited slightly more by Dave Walden.

Notes and References

1. A 1994 attempt to provide a complete bibliography of *published papers* from BBN in the networking area resulted in a 39-page document (*ARPANET and Internet: A Bibliography of BBN Papers*, compiled by A. McKenzie, D. Walden, and F. Heart, Bolt Beranek and Newman Inc., Cambridge, MA, 1994). This internal BBN document was distributed to the attendees of a party marking the 25th anniversary of the beginning of the ARPANET, organized by BBN and held at the Copley Plaza Hotel in downtown Boston. No doubt many more networking papers by BBNers have been published in the last 15 years. A version of the 1994 bibliography with different page breaks is available at www.walden-family.com/bbn/anin-biblio.pdf

2. K. Hafner and M. Lyon, *Where Wizards Stay Up Late*, Simon and Schuster, New York, 1996.

3. P. Salus, *Casting the Interne*t, Addison-Wesley, Reading, MA, 1995.

4. J. Abbate, *Inventing the Internet*, MIT Press, Cambridge, MA, 1999.

5. During some years ARPA has been known as DARPA — Defense Advanced Research Projects Agency. We use ARPA throughout this chapter.

6. "Request for Quotation, Department of the Army Defense Supply Service–Washington, 1968 July 29"; a procurement managed on behalf of the Advanced Research Projects Agency.

7. Heart, Kahn, Ornstein, and Crowther made the trip.

8. SRI International was founded as the Stanford Research Institute.

9. F. E. Heart, R. E. Kahn, S. M. Ornstein, W. R. Crowther, and D. C. Walden, "The Interface Message Processor for the ARPA Computer Network," *AFIPS Spring Joint Computer Conference Proceedings*, vol. 36, AFIPS Press, 1970, pp. 551–567.

10. Doing new releases across the network became possible once an IMP at BBN was on the network. A new release was loaded into the BBN IMP; then each neighboring IMP of the BBN IMP was instructed to reload from the BBN IMP, and then neighbors of those IMPs reloaded, and so on across the network.

11. Steve Crocker, then a graduate student at UCLA, has described UCLA as in a near "panic" to prepare when they discovered the IMP was going to be delivered on time.

12. E-mail of July 18, 2006.

13. A. A. McKenzie, B. P. Cosell, J. M. McQuillan, and M. J. Thrope, "The Network Control Center for the ARPANET," *Proceedings of the First International Conference on Computer Communications*, ed. S. Winkler, Washington, DC, 1972, pp. 185–191.

14. R. E. Kahn and W. R. Crowther, "Flow Control in a Resource-Sharing Computer Network," *IEEE Transactions on Communications*, vol. COM-20, no. 3, pp. 536–546.

15. Ornstein et al., Users Guide to the Terminal IMP, BBN Report 2183, July 1972.

16. BBN was awarded the IEEE 1999 Corporate Innovation Recognition award for its development of the IMP and Terminal IMP (TIP).

17. See section 17.4 of this chapter, "Radio and Wireless."

18. See section 21.6, page 534.

19. J. M. McQuillan, I. Richer, and E. C. Rosen, "The New Routing Algorithm for the ARPANET," *IEEE Transactions on Communications*, vol. COM-28, no. 5, May 1980, pp. 711–719.

20. D. G. Bobrow, J. D. Burchfiel, D. L. Murphy, and R. S. Tomlinson, "TENEX, a Paged Time Sharing System for the PDP-10," *Proceedings of the Third ACM Symposium on Operating Systems Principles*, ACM Press, 1971, pp. 135–143.

21. C. S. Carr, S. D. Crocker, and V. G. Cerf, "Host-Host Communication Protocol in the ARPA Network," *AFIPS Spring Joint Computer Conference Proceedings*, vol. 36, AFIPS Press, 1970, pp. 589–597.

22. A. McKenzie, Initial Connection Protocol, Request for Comments 93 (RFC-93), January 1971; `http://www.rfc-editor.org/rfc/rfc93.txt`

23. J. Davidson, W. Hathaway, N. Mimno, J. Postel, R. Thomas, and D. Walden, "The ARPANET TELNET Protocol: Its Purpose, Principles, Implementation, and Impact on Host Operating System Design," *Proceedings of the ACM/IEEE 5th Data Communications Symposium*, 1977, pp. 4-10 to 4-18.

24. B. P. Cosell and D. C. Walden, "Development of TELNET's Negotiated Options," *IEEE Annals of the History of Computing*, vol. 25, no. 2, 2003, pp. 80–82.

25. Alex McKenzie (for example, RFC-281 and RFC-454) and Nancy Neigus (for example, RFC-542) of BBN were particularly active with FTP.

26. BBN's role in the development of e-mail is described in Chapter 19 and in Craig Partridge's broader (than BBN) paper on the technical history of Internet e-mail cited at the beginning of Chapter 19.

27. *Where Wizards Stay Up Late*, pp. 176–186.

28. See also `http://video.google.com/videoplay?docid=4989933629762859961` (posted on the Internet as of February 2008), a video undoubtedly made around the same time as the conference and which justifies the concept of packet switching.

29. E. W. Wolf, "An Advance Computer Communications Network," AIAA Computer Network Systems Conference, Huntsville, Alabama, April 1973.

30. N. J. Liaaen and D. C. Walden, "Remembering the LFK Network," *IEEE Annals of Computing*, vol. 24, no. 3, July–September 2002, pp. 79–81.

31. L. Pouzin, "Presentation and Major Design Aspects of the CYCLADES Computer Network," *Computer Communication Networks*, eds. R. Grimsdale and F. Kuo, Nordhoff Publishing Co., Leyden, The Netherlands, 1975 (reprint of a 1973 paper).

32. "Experiences in Building, Operating, and Using the ARPA Network," *Proceedings of the 2nd USA-Japan Computer Conference*, Tokyo, August 1975, pp. 453–458.

33. Kirstein is well known in the history of the Internet, having been involved, for instance, in some of the earliest experiments with connecting networks together. He was recipient of the 1999 ACM SIGCOMM Award and the Internet Society's 2003 Jonathan B. Postel Service Award, and he has received many other honors for his work on the Internet.

34. Chapter 6 of this book.

35. Inside BBN, Bob Hinden strongly argued that BBN should suggest to DCA to use Ethernet rather than X.25 as the interface standard for ARPANET.

36. Some of the story of the transition from the ARPANET to DDN and the Internet is also told by A. A. McKenzie and D. C. Walden, "The ARPANET, The Defense Data Network, and The Internet," *Encyclopedia of Telecommunications*, vol. 1, Marcel Dekker, 1991, pp. 341–346.

37. Alexander McKenzie, "NWG and the Conception of the Internet: An Eyewitness Account," *IEEE Annals of the History of Computing*, vol. 33, no. 1, pp. 66–71.

38. V. Cerf and R. Kahn, "A Protocol for Packet Network Interconnection," *IEEE Transactions on Communications*, vol. COM-22, 1974, pp. 637–684.

39. Canada, represented by Bell Canada, was an exception.

40. S. C. Butterfield, R. D. Rettberg, and D. C. Walden, "The Satellite IMP for the ARPA Network," *Proceedings of the Seventh Annual Hawaii International Conference on System Sciences*, IEEE CS Press, 1974, pp. 70–73.

41. J. Burchfiel, R. S. Tomlinson, and M. Beeler, "Functions and Structure of a Packet Radio Station," *AFIPS Conference Proceedings*, vol. 44, AFIPS Press, 1975, pp. 245–251.

42. See "TCP Research" below.

43. J. Abbate, *Inventing the Internet*, p. 131.

44. Exactly how the splitting of TCP and IP was conceived is not documented. It apparently happened in a hallway during a Network Working Group meeting at USC ISI sometime in 1977, according to a personal communication from David Reed on July 14, 2004.

45. V. G. Cerf, A. A. McKenzie, R. A. Scantlebury, and H. Zimmerman, "Proposal for an Internetwork End-to-End Protocol," *ACM SIGCOMM Computer Communications Review*, vol. 6, no. 1, 1976, pp. 63–89.

46. There was a sense that the work on TCP was an implicit criticism of the existing work done on ARPANET, which provided a service more like a virtual circuit than a string of datagrams.

47. BCPL is a programming language developed in the United Kingdom by Martin Richards. It was a precursor to the C programming language, and it was widely used at BBN and elsewhere in the ARPA research community in the early days of the ARPANET.

48. TENEX was the first major operating system to use paging and supported innovative features such as: clear distinction between operating system and applications, command line completion, and file versioning. TENEX was created and maintained by BBN, and having easy access to the source of a (widely used) operating system was an important resource for BBN's networking research. The development of TENEX is described in Chapter 21.

49. R. S. Tomlinson, "Selecting Sequence Numbers," *Proceedings of the ACM SIGCOMM/SIGOPS Interprocess Communications Workshop*, ed. D. Walden, ACM Press, 1975, pp. 11–26.

50. For a good discussion of the sequence number problem as applied to routing protocols, see R. Perlman, *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*, 2nd ed., Addison-Wesley, Reading, MA, 1999, pp. 310–317.

51. As part of this work, Plummer wrote several influential notes on TCP implementation. However, oddly enough, he is best remembered for his vigorous but unsuccessful attempt to keep Rubber EOLs in the TCP specification.

52. This last idea was borrowed from work by Tom Blumer on OSI protocol implementation; see "Open Systems Interconnection (OSI)" below.

53. R. Gurwitz and R. Walsh,"Converting the BBN TCP/IP to 4.2BSD," *Proceedings 1984 Summer USENIX Conference*, Usenix Assoc., 1984, pp. 52–61.

54. While this is Charlie Lynn's only mention in this chapter, his contributions were far bigger. Charlie was one of BBN's best implementers for three decades. He specialized in finding ways to implement complex networked systems and served as a mentor to quite a few of the people mentioned in this chapter. He passed away unexpectedly in 2004.

55. Now the National Institute of Standards and Technology (NIST).

56. D. Deutsch, R. Resnick, and J. Vittal, Specification of a Draft Message Format Standard, BBN Report 4486, 1980.

57. R. D. Rettberg and D. C. Walden, "Gateway Design for Computer Network Interconnection," *Proceedings of Eurocomp (The European Computing Conference on Communications Networks) 1975*, Online Conferences Ltd., 1975, pp. 113–128.

58. M. Beeler, J. Burchfiel, R. Rettberg, V. Strazisar, and D. Walden, "Gateway Design for Computer Network Interconnection," invited presentation (given by Strazisar), *Proceedings of AFIPS 1976 National Computer Conference and Exposition*, AFIPS Press, 1976.

59. V. Strazisar,"Gateway Routing: An Implementation Specification," Internet Engineering Note 30 (IEN-30), April 1978.

60. V. Strazisar, "How to Build a Gateway," Internet Engineering Note 109 (IEN-109), August 1979.

61. R. Hinden and A. Sheltzer, "The DARPA Internet Gateway," Request for Comments 823 (RFC-823), September 1982.

62. By 1983, Mike Brescia had written a memo known informally as "Mike's Instructions For

Building Your Own Gateway" which described in detail what type of LSI-11 to order, what network cards to order, and how to get a software tape from Mike.

63. Ginny Travers, Bob Hinden, and Mike Brescia received the 2000 IEEE Internet Award "for pioneering work in the development of the first Internet routers."

64. BBN's parallel-processing computer research and development from the Pluribus through the Butterfly and beyond is described in Chapter 21.

65. Described in T. Mallory, "SPF Routing in the Butterfly Gateway," *Proceedings of the April 22–24, 1987 Internet Engineering Task Force Meeting (Sixth IETF)*, ed., P. Gross, `http://www3.ietf.org/proceedings/prior29/IETF06.pdf`

66. Randy Rettberg felt that it would be efficient to combine the boot load device and the console into a single platform and proposed using small Macintosh personal computers for this function. One result was that the Butterfly loaded its boot image over the Mac's very slow serial port.

67. Bob Hinden led the team that designed the initial prototype, code-named "Emerald."

68. C. Partridge et al., "A Fifty Gigabit Per Second IP Router," *IEEE/ACM Transactions on Networking*, vol. 6, no. 3, 1998, pp. 237–248.

69. Interface Message Processors for the ARPA Computer Network: Quarterly Technical Report No. 1, BBN Report 1783, March 31, 1969, pp. 10–14.

70. D. Walden, "The Interface Message Processor, Its Algorithms, and Their Implementation," invited lecture, Journees D'Etude Reseaux de Calculateurs, Association Francaise pour la Cybernetique Economique et Technique, Paris, 1972, `http://walden-family.com/public/1972-afcet-paris.pdf`

71. D. Walden, "Routing (a memorandum)," *Proceedings of 1974 International Seminar on Performance Evaluation of Data Processing Systems*, Weizmann Institute of Science, Rehovot, Israel, 1974, pp. 429–433.

72. J. M. McQuillan and D. C. Walden, "The ARPA Network Design Decisions," *Computer Networks*, vol. 1, no. 5, 1977, pp. 243–289.

73. It is also known as the "distributed Bellman-Ford" algorithm, although BBN's parallel algorithm for ARPANET had little similarity to Bellman's or Ford's nonparallel algorithms (D. Walden, "The Bellman-Ford Algorithm and Distributed Bellman-Ford," `http://www.walden-family.com/public/bf-history.pdf`).

74. Many of these weaknesses were foreseen by team members Will Crowther and Bob Kahn, but the initial algorithms were probably the best the team could devise and implement within the time available for initial network deployment (see "ARPANET Grows" in section 17.1, "ARPANET").

75. J. M. McQuillan, *Adaptive Routing Algorithms for Distributed Computer Networks* — McQuillan's Harvard PhD thesis, which was reprinted as BBN Report 2831, 1974.

76. J. M. McQuillan, I. Richer, and E. C. Rosen, "The New Routing Algorithm for the ARPANET," *IEEE Transactions on Communications*, vol. COM-28, no. 5, 1980, pp. 711–719.

77. R. Perlman, *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*, 2nd ed., Addison-Wesley, Reading, MA, 1999.

78. J. M. McQuillan e-mail of April 22, 2003, to David Walden.

79. John M. McQuillan, "The Birth of Link-State Routing," *IEEE Annals of the History of Computing*, vol. 31, no. 1, January–March 2009, pp. 68–71.

80. Robert Hinden and Alan Sheltzer, The DARPA Internet Gateway, Request for Comments 823 (RFC-823), September 1982.

81. E. C. Rosen, "Exterior Gateway Protocol," Request for Comments 827 (RFC-827), October

1982. Much of the credit for EGP is shared with David Mills, then of Linkabit Corporation. Mills took Rosen's somewhat sketchy specification of EGP and made it concrete and stable (cf. D.L. Mills, "Exterior Gateway Protocol Formal Specification," Request for Comments 904 (RFC-904), April 1984).

82. N. Abramson, "The ALOHA System — Another Alternative for Computer Communications," *AFIPS Fall Joint Computer Conference*, AFIPS Press, 1970.

83. L. Roberts, "ALOHA Packet System with and without Slots and Capture," ACM SIGCOMM *Computer Communications Review*„ vol. 5, no. 2 April 1975; R. Metcalfe, "Steady State Analysis of a Slotted and Controlled Aloha System with Blocking," same journal issue (Metcalfe's working note was written in response to Roberts's).

84. W.R. Crowther, R.D. Rettberg, F.E. Heart, S.M. Ornstein, and D.C. Walden, "A System for Broadcast Communication: Reservation-ALOHA," *Proceedings of the Sixth Hawaii International Conference on Information and System Sciences*, January 1973, pp. 371–374; this paper was written after the original versions of the Roberts and Metcalfe papers listed in the prior note.

85. Among other BBN people who participated were Nils Liaaen, Bob Bressler, Robert Weissler, Nai-Ting Hsu, Tony Lake, and Jane Barnett. Binder and Hsu both joined Linkabit some years later.

86. I.M. Jacobs, R. Binder, and E.V. Hoversten, "General Purpose Packet Satellite Networks," *Proceedings of the IEEE*, vol. 66, no. 11, pp. 1448–1467.

87. For more about this group's activities, see Chapter 14.

88. The Pluribus multiprocessor is described in more detail in Chapter 21.

89. G. Falk, J.S. Groff, W.C. Milliken, M. Nodine, S. Blumenthal, and W. Edmond, "Integration of Voice and Data in the Wideband Packet Satellite Network," *IEEE Journal on Selected Areas in Communications*, vol. SAC-1, no. 6, 1983, pp. 1076–1083.

90. The Butterfly multiprocessor is described in more detail in Chapter 21.

91. W. Edmond, S. Blumenthal, A. Echenique, S. Storch, T. Calderwood, and T. Rees, "The Butterfly Satellite IMP for the Wideband Packet Satellite Network," *Proceedings of ACM SIGCOMM 1986*, ACM Press, 1986, pp. 194–203.

92. For more about Diamond, see Chapter 18.

93. For more about SIMNET, see Chapter 20.

94. M. Bergamo, ACTS Gigabit Satellite Network Study: Satellite Beam Switched TDMA Networking and Support for SONET Interfaces, BBN Report 7574, March 29, 1991.

95. M. Bergamo, "Network Architecture and SONET Services in the NASA/DARPA Gigabit Satellite Network using NASA's Advanced Communications Technology Satellite," *Proceedings of the 15th AIAA International Communications Satellite Systems Conference*, AIAA, 1994, pp. 208–216.

96. M. Bergamo and D. Hoder, "Gigabit Satellite Network for NASA's Advanced Communications Technology Satellite (ACTS)," *International Journal of Satellite Communications*, vol. 14, no. 3, 1996, pp. 161–173.

97. R. Ramanathan and J. Redi, "A Brief Overview of Ad Hoc Networks: Challenges and Directions," *IEEE Communications Magazine*, 50th anniversary commemorative issue, May 2002, pp. 20–22.

98. R.E. Kahn, S.A. Gronemeyer, J. Burchfiel, and R.C. Junzelman, "Advances in Packet Radio Technology," *Proceedings of the IEEE*, vol. 66, no. 11, November 1978, pp. 1468–1496.

99. Others who participated in the project at BBN included Jil Westcott, Ray Tomlinson, Radia Perlman, Don Allen, Mike Beeler, Ginny Travers, and Greg Lauer.

100. E-mail of September 2, 2003.

101. While an inter-divisional attitude of not-invented-here was perhaps part of the reason for

BBN Communications Corporation eschewing the network management system Jil describes, there were also other network management projects in the other computer division and within the Communications Corporation.

102. J. Jubin and J. D. Tornow, "The DARPA Packet Radio Network Protocols," *Proceedings of the IEEE*, vol. 75, no. 1, 1987, pp. 21–32. Other participants playing long-term roles in this program included Jil Westcott and Greg Lauer.

103. NTDR was the first "real-life" ad hoc network; it was used by the Fourth Infantry Division in Operation Desert Storm.

104. S. Ramanathan and M. Steenstrup, "Hierarchically-Organized, Multihop Mobile Networks for Multimedia Support," *ACM/Baltzer Mobile Networks and Applications*, vol. 3, no. 1, pp. 101–119.

105. K. Kasera and S. Ramanathan, "A Location Management Protocol for Hierarchically Organized Multihop Mobile Networks," *Proceedings of the IEEE International Conference on Universal Personal Computing (ICUPC 97)*, IEEE Press, 1997.

106. The last several paragraphs of this subsection were updated based on an e-mail exchange involving Craig Partridge, Jason Redi, and Jim Freebersyser, March 27–31, 2010.

107. R. Ramanathan, "On the Performance of Ad Hoc Networks Using Beamforming Antennas," *Proceedings of ACM MobiHoc 2001*, ACM Press, 2001, pp. 95–105.

108. J. Redi and R. Ramanathan, "Utilizing Directional Antennas for Ad Hoc Networks," *Proceedings of IEEE Milcom*, IEEE Press, 2002.

109. J. Redi and B. Welsh, "Energy-Conservation for Tactical Robot Networks," *Proceedings of IEEE Milcom*, IEEE Press, 1999, pp. 1429–1433.

110. J. Redi, S. Kolek, K. Manning, C. Partridge, R. Rosales-Hain, R. Ramanathan and I. Castineyra, "JAVeLEN: An Ultra-Low Energy Ad hoc Wireless Network," *Ad Hoc Networks*, vol. 6, no. 1, January 2008, pp. 108–126.

111. R. Ramanathan and R. Hain, "Topology Control of Multihop Radio Networks Using Transmit Power Adjustment," *Proceedings of IEEE Infocom*, IEEE Press, 2000, pp. 404–413.

112. R. Ramanathan and R. Hain, "An Ad Hoc Wireless Testbed for Scalable, Adaptive QoS Support," *Proceedings of the IEEE Wireless Communication and Networking Conference*, vol. 3, IEEE Press, 2000, pp. 998–1002.

113. R. Ramanathan, "Making Ad Hoc Networks Density Adaptive," *Proceedings of IEEE Milcom*, IEEE Press, 2001, pp. 957–961.

114. C. Santivanez, R. Ramanathan, and I. Stavrakakis, "Making Link-State Routing Scale for Ad Hoc Networks," *Proceedings of ACM MobiHoc 2001*, ACM Press, 2001, pp. 22–32.

115. J. Redi, R. Ramanathan, "Utilizing Directional Antennas for Ad Hoc Networks," *Proc. IEEE Milcom 2002*.

116. L. Dai, P. Basu, and J. Redi, "An Energy Efficient and Accurate Slot Synchronization Scheme for Wireless Sensor Networks," *Proceedings of The Third IEEE/Createnet International Conference on Broadband Communications, Networks and Systems* (Broadnets 2006), San Jose, CA, October 2006.

117. C. Santivanez, B. McDonald, I. Stavrakakis, R. Ramanathan, "On the Scalability of Ad Hoc Routing Protocols," *Proceedings of IEEE Infocom*, 2002.

118. Some other relevant papers are: S. Ramanathan, "A Unified Framework and Algorithm for Channel Assignment in Wireless Networks,"*Wireless Networks* 5 (1999), pp. 81–94; S. Ramanathan and M. Steenstrup, "A Survey of Routing Techniques for Mobile Communications Networks," *ACM/Baltzer Mobile Networks and Applications*, vol. 1, no. 2, pp. 89–103; S. Ramanathan, "Multicast Tree Generation in Networks with Asymmetric Links," *IEEE/ACM Transactions on Networking*, vol. 4, no. 4, pp. 558–568.

119. E-mail of June 19, 1996, in a discussion group thread on "network layer encryption history and prior art," at `http://www.sandelman.ottawa.on.ca/ipsec/1996/06/msg00050.html`

120. E-mail of August 24, 2003.

121. See section 17.1, "ARPANET Influences and Spinoffs."

122. Steve Kent also notes, "In the early-80s, BBN developed the IPLI, a successor to the PLI, updated to use TCP/IP, newer COMSEC technology, but still manually keyed. The IPLI was a backup program, funded by ARPA and DCA, in case the more ambitious (multi-level secure) Blacker program was delayed (which it was) and caused programmatic problems for the newly inaugrated Defense Data Network (DDN). IPLIs also were designed for a tactical environment, for use with ARPA low cost packet radios. A small number of IPLIs were delivered in the mid-80s, but never saw real deployment."

123. Because of the nature of the area and clients for the work, there are not as many publications in the security area as in many other areas. Some representative publications in the security area follow: S. Kent, Evaluation of DES on PLATFORM/OPSN, BBN Report 4929, March 1982; S. Kent, and J. Linn, PLATFORM/OPSN Privacy System: Preliminary Software Design, BBN Report 5099, August 1982; S. Kent, J. Linn, and G. Ruth,PLATFORM/OPSN Privacy System: Overall Design, BBN Report 5091, July 1982; J. Herman, S. Kent, and P. Sevcik, "Personal Authentication System for Access Control to the Defense Data Network," *Proceedings of the 15th Annual IEEE Electronics and Aerospace Systems Conference (EASCON)*, September 1982; G. Kajos, S. Kent, K. Pogran, and J. Walters, Compartmented Local Area Network Study for the Electronic Security Command, BBN Report 5379, September 1983; S. Kent, J. Linn, and Z. Opalka, COINS-SCINET Interim Secure Gateway: Preliminary Design Analysis, BBN Report 6110, December 1985; R. Hinden, S. Kent, J. Linn, and Z. Opalka, Preliminary Design Specifications for the Gateways Between the COINS and DoDIIS Networks, BBN Report 6407, November 1986; J. Linn and S. Kent, "Privacy for DARPA Internet Mail," *Proceedings of 12th National Computer Security Conference*, October 1989; S. Kent and K. Rossen, "E-Mail Privacy for the Internet," *Business Communications Review* vol. 20, no. 1, January 1990; S. Kent, "Internet Privacy Enhanced Mail," *Communications of the ACM* vol. no. 6, July 1993, pp. 48–60; S. Kent, "An Overview of Internet Privacy Enhanced Mail," *Proceedings of INET92*, June 1992, pp. 217–227; S. Kent, P. Helinek, D. Ellis, K. Sirois, and N. Yuan, Internet Routing Infrastructure Security Countermeasure Design, BBN Report 8173, July 1996; S. Kent, K. Sirois, and D. Ellis, Internet Routing Infrastructure Security Requirements Analysis, BBN Report 8141, January 1996; J. Zao, S. Kent, J. Gahm, et al., "A Public-Key based Secure Mobile IP," *Wireless Networks*, vol. 5 (1999); C. Lynn and K. Seo, "Design and Analysis of the Secure Border Gateway Protocol (S-BGP)," *IEEE DISCEX Conference*, January 2000; C. Lynn, J. Mikkelson, and K. Seo, "Secure Border Gateway Protocol (S-BGP) — Real World Performance and Deployment Issues," *Proceedings of the Symposium on Network and Distributed System Security*, February 2000; "Secure Boarder Gateway Protocol (S-BGP)," with S. Kent, C. Lynn and K. Seo, *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, April 2000; L. Sanchez, W. Milliken, A. Snoeren, F. Tchakountio, C. Jones, S. Kent, C. Partridge, and W. Strayer,"Hardware Support for a Hash-Based IP Traceback," *DARPA Information Survivability Conference and Exposition (DISCEX) II*, Anaheim, CA, June 12–14,2001, vol. 2, pp. 146–152; A. Snoeren, C. Partridge, L. Sanchez, C.. Jones, F. Tchakountio, S. Kent, and W. Strayer, 'Hash-Based IP Traceback," *Proceedings of the ACM/SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '01)*, San Diego, CA, August 2001, pp. 3–14,.

124. Chip Elliott, "Building the Quantum Network," *New Journal of Physics*, vol. 4, 2002, pp. 46.1–46.12.

125. This paragraph on quantum methods of security was provided by participant Chip Elliott in an e-mail of November 28, 2006.

126. One of the authors of this chapter, Craig Partridge, had a characteristic encounter with Brescia. Partridge was debugging a new network management protocol implementation and, to see if he could use it to retrieve data from a local router, sent a few packets to the router. On getting no reply, he reread his code for bugs, then tried again. Still no answer from the router, but the phone rang. It was Brescia calling to say that Partridge's implementation had two bytes reversed in the protocol header and to please fix it.

127. D. Comer, "The Computer Science Research Network CSNET: A History and Status Report," *Communications of the ACM*, vol. 26, no. 10, pp. 747–753.

128. In the 1980s, there were several competing e-mail networks and getting the e-mail between networks required considerable skill (see J. S. Quartermann, *The Matrix*, Digital Press, 1990). Charlotte Mooers of BBN, CSNET's "postmistress" became internationally known for her skill in getting e-mail to the right place.

129. C. Partridge, "Mail Routing and the Domain Name System," Request for Comments 974 (RFC-974), January 1986.

130. L. Lanzillo and C. Partridge, "Implementation of Dial-up IP for UNIX Systems," *Proc. 1989 Winter USENIX Conference*, Usenix Assoc., pp. 201–208.

131. One of Rugo's marketing challenges was finding out who in a high-tech company was running the company's network. This was before the days of chief technology officers (CTOs). So Rugo took to calling each company's CEO, on the assumption that he'd get the CEO's assistant, who could tell him who ran the company's network. Some of his most memorable marketing calls occurred when he actually got put directly through to the CEO of a major high-tech company.

132. NEARNET chose to spend its NSF start-up money on capital equipment, whereas most regional networks chose instead to use the money to reduce what they charged customers. When the money ran out, many regionals were left with an infrastructure in need of upgrading and customers suffering sticker shock.

133. Probably, in part, because some key operations people were at Planet, including Dan Long; John Curran (ex-CSNET, and Planet's CTO); Mike Brescia; and Steve Blumenthal (who was manager of the group that ran the Wideband network and several other experimental networks and who succeeded Curran as Planet's CTO).

134. D. Greenfield, "North American ISP Survey: Looking for Number Two," *Network Magazine*, September 2002, `http://www.itarchitect.com/article/NMG20020822S0001`.

135. BBN's 1996 10K filing with the Securities and Exchange Commission (SEC) shows that Planet's revenues (closely tied to customers and bandwidth) more than doubled between June 1994 and June 1995, and more than quadrupled between June 1995 and June 1996.

136. A comment on the economics of hypergrowth in the 1990s may be useful. Deploying new networking equipment and circuits typically took 6 to 12 months from initial ordering until completed installation. So an ISP had to guess how big it would be a year in advance. The penalty for underestimating was crushing — loss of market share, reduced revenue stream, devalued stock. The penalty for overestimating was mild — the excess capacity would be consumed in the next year, provided hypergrowth continued. BBN's two biggest competitors in this period were Uunet and PSI. PSI was the most aggressive in pursuing a "grow at all costs" policy, while Uunet (until it was acquired by WorldCom) was the most fiscally conservative of the three competitors.

Chapter 18

Distributed Computing at BBN

Network Computing Software Infrastructure and Distributed Applications from 1970-1995

Richard Schantz

In this article we review highlights of the history of BBN and distributed computing from a number of different perspectives: early investigations, distributed systems infrastructure (later to be more commonly known as Middleware), and network-centric applications which utilized the emerging distributed systems capabilities in new and innovative ways, with lasting impact.

18.1 Introduction

The innovations being pursued in the area of packet switching and inter-computer communications (see Chapter 17 on networking) spawned a simultaneous interest and investigation into how to utilize this emerging new data oriented communication capability. To complement its activities in pioneering network communications, from the earliest conception of the ARPANET and continuing to this day, BBN initiated and participated in a number of the leading edge, innovative activities aimed at delivering value from the increasing interconnectivity through network centric applications and by providing the network centric infrastructure needed to more easily build these types of applications (see also Chapter 19 on email and 20 relating to distributed simulation). We use the term "Distributed Computing" to loosely tie together these activities covering both advanced, higher levels of multi-host[1] infrastructure and the innovative distributed applications themselves, and to distinguish it from the lower level capabilities (e.g., network communications) discussed in Chapter 17. While advances in the infrastructure needed to support distributed applications have been steadily evolving, both at BBN and elsewhere, since the inception of the network, the advent of significant new application areas is somewhat more discrete. Three of these application areas, e-mail in the 1970s (Chapter 19) as the first major new successful network application, distributed interactive simulation in the 1980s (Chapter 20[2]), and multimedia applications spanning the 1980s but effectively emerging in the 1990s, got their start or were early trailblazing activities within BBN R&D projects, and required significant redirection of the needed network support. Like the distributed systems infrastructure support work, each of these application areas which had significant roots in the early ARPANET/Internet capabilities, now have whole industries which are continuing to push these areas forward to this day.

By the early 1970s, BBN had successfully developed and quickly expanded the first network of packet switches. But it took more than Interface Message Processors (IMPs)

(Chapter 17) to make the emerging network truly useful. The initial goal of designing, implementing, and fielding the ARPANET[3] generated new interest in the kinds of protocols and distributed computing technology that could further enhance the utility of the experimental network. In its initial conception, the uses for the network were vague and speculative. By 1975, protocol investigation within DARPA had already evolved to the concept of a network of networks — which ultimately became the Internet — and to TCP/IP as a means for universal interconnection. Simultaneous with this evolution of the network communication, BBN, as part of both DARPA's research community and the university computer science research community at large, had already begun speculating on and investigating the impact of computer-to-computer communication capabilities and resource sharing on new applications such as distributed air traffic control, unified medical records across hospitals, distributed file systems, and on the operating systems software needed to deliver communication support services to those applications.

Early work in these areas proved conclusively that such applications were complex, difficult to build and relied on common availability and interpretation of capabilities across the nodes of the network. This led to two parallel tracks of continued investigation and development. One concentrated on the direct implementation of a very focused set of immediately useable end applications, starting with those of immediate utility to the developers themselves such as telnet, ftp, and e-mail, while the other focused on additional infrastructure in the form of network or distributed operating systems to enable more general-purpose applications to be developed faster and easier using wide area communications capabilities. The latter activity spawned what has come to be known as "middleware," because it was strategically placed between the network and the application to provide a richer, simpler network environment for a wide variety of uses. Later, more sophisticated and complex applications, focusing on providing value to "plain old users" of the interconnected capability, in areas such as distributed simulation and training, and tools for multi-media collaboration, became important drivers for new network-centric capabilities. In each case, BBN scientists and engineers were there at the beginning to develop the early prototypes and establish the technical agendas which continue to be pursued and refined by the industry at large.

In the remainder of this chapter of the history of computing at BBN, we will try to maintain a chronological perspective on the activities. However, there often were a number of parallel and independent activities evolving simultaneously. So to provide a smoother presentation, we will often pursue a topic through many years of its own particular evolution, and then return to an earlier epoch to repeat those same years from a different vantage point. In that way we hope to obtain the right mix of chronological relationships with continuity of ideas and threads of investigation and development.

18.2 Early investigations: 1970-1973

The birth and accessibility of the emerging multi-node communications capability, which was intended to be attached to any number of computing installations, led immediately to a number of threads of investigation and experimentation, all of which centered on the question "now that we have this interconnected laboratory, what can we do with it and how can we use it?" Among these threads were issues of how to make the networking capability available to users and application developers through the operating systems that already ran the host computers being connected to the network, how to share resources using this medium and what sort of resources were effective for sharing, and what applications could be conceived and built that would make immediate use of the emerging capabilities, for ourselves as well as for others.

For BBN scientists, the first order of business was connecting the TENEX operating system, the time sharing system built earlier by BBN to support virtual memory and large address space applications for the PDP-10 family of computers, to the network.[4] This activity, described in Chapter 21, provided flexible and general access to network services through an augmented file system interface. ARPANET-enabled TENEX was made available to the ARPA research community by 1971, and in short order became the "standard" ARPA research site computational engine. In contrast, a number of the other operating system network integration efforts were not nearly as flexible or general purpose, sometimes limiting the capability of developing networked applications to a single application or use. This issue of how to reflect the networking capabilities to applications is an area of continuing innovation and experimentation to this day, as all of the opportunities and complexities associated with more and more sophisticated use of the interconnected computing capabilities has continued to emerge. The community process that spawned the initial high-level communications protocols for host to host stream connections (NCP), also developed protocols for two "quasi-applications," of immediate use and utility to those building the network, that could promote resource sharing. These are being called "quasi-applications" because they were not really applications at all (except to the developers of the network itself); rather they were the first instances of value-added services: Telnet, to enable connecting a terminal device to operate a remote computer (e.g., for remote login), and File Transfer Protocol (FTP) to move files in various formats from one host computer to another. Using those two primitive tools, one could make effective use of a remote computer, provided you knew enough of the details about using the remote host, had an account on it, and didn't mind the delayed responses. BBN scientists played significant roles in developing and enhancing these consensus based protocols, and especially in providing implementations of them for the TENEX operating system.[5]

Beyond these community-oriented activities, a few early BBN centric experimental activities set in motion threads of investigation which were to have considerable impact on things to come. First, in 1971 Ray Tomlinson, who earlier was instrumental in developing the ARPANET host software for TENEX, experimented with hooking up the network capabilities available through TENEX to programs which could send and receive text messages across the network, much like existing timesharing "mail" programs were already doing for users of a single timesharing host. This created the first interhost e-mail capability and the first new use of the communication capability beyond remotely using another machine. This activity laid the foundation, the architectural framework and some of the conventions for networked e-mail as the first network centric application with potential applicability outside of the computer science community which spawned the networking innovations. It developed the first network e-mail sending program (SNDMSG), established the @-sign separator for `name@host` mail addressing, and established the business memo format (To, Subject, From, Date, CC). It also set the direction for decades of work in defining, developing and refining such a facility. This innovation has been so significant that it has worked its way into the popular media, and has been recounted in a number of publications. The broader story of the development of e-mail in the early years is already well told in chapter 7 of Hafner and Lyon's book, *Where Wizards Stay Up Late*.[6]

At about the same time as the e-mail experimentation, Bob Thomas, who had earlier joined the BBN TENEX group after getting his PhD degree from MIT, was using the access to the network to develop a prototype for a multi-computer air traffic control application (Multi-computer Route Oriented Simulation System, or McROSS). This application instantiated different air traffic route controllers on different nodes, and developed the interoperating software modules for controlling air traffic and handing off aircraft controlled by one node to another. While this was not a serious application (it was a simulation of how such a facility might operate) it did demonstrate the utility, capability and issues involved with machine to machine interaction in a pure end use application context (not remotely connecting terminals or passing files from one computer to another). Perhaps more importantly, it developed ideas for and implementation of software packages as utilities whose purpose it was to make it easier to provide commonly used functions for network computing. In particular, it focused on the capabilities that an operating system process might need to more easily build networked computing applications, such as air traffic control, with dynamic interactions between the parts of the distributed application. The experience with handing off a piece of computation from one machine to another led to experimentation with and demonstration of a program called "Creeper" which became the world's first computer worm: a computation that used the network to recreate itself on another node, and spread from node to node.

Bob Thomas remembers:[7]

> I joined BBN in '71 after spending several stressful years narrowly focused on completing my thesis. I joined a group that had just been awarded an ARPA contract for research on distributed computing. My assignment, as I understood it at the time, was to address the question "How can the computers (being) connected to the ARPANET use it?," and to provide answers in the form of working prototype applications and systems. Remote terminal access and file transfer were obvious and visionaries like Englebart, Roberts, Kahn and others had ideas but the space, particularly the technical problems in supporting network applications, was largely unexplored. The assignment was essentially to play in a new large sandbox with a group of very bright people; going to work each day was a tremendous high; just about everything I worked on was new and interesting.
>
> The group that I joined had developed a simulation system for studying the possible automation of air traffic control procedures. The system, called ROSS (for Route Oriented Simulation System), ran on a DEC PDP-10 computer, and simulated aircraft in an airspace.
>
> We had the idea that we could build a distributed version of this system that could run on different computers and use the ARPANET to support interaction among its distributed parts. The goal was to investigate distributed computing and networking issues. The goals of the air traffic control simulation studies could just as well been met with the existing non-distributed version of ROSS.
>
> I built the distributed version of the simulation system and named it McROSS (for Multi-computer Route Oriented Simulation System).[8] You could use McROSS to simulate several adjacent air spaces and the air traffic within those air spaces, where the simulation for each air space could run on a different computer. In addition, simulated aircraft could fly out of one airspace and into another, in effect out of one computer on the ARPANET and into another as responsibility for simulating the aircraft passed with the aircraft from computer to computer. I also built a companion program that could "attach" to one of the simulated airspaces and request aircraft data (i.e., altitude, position, velocity, etc.) for the purpose of displaying the aircraft, much like a typical air traffic control display.
>
> I then got the idea that it might be useful to be able to move a part of a simulation from one computer to another without interrupting the progress of the on-going simulation — that is, to move the simulation of an airspace to another computer in a way that retains its connectivity with the other parts of the simulation and that ensures that all of the simulated aircraft continue to be correctly simulated, including any that might be in the process of being handed off to another simulation component. Doing this would require the part that moves to pick itself up with all of its aircraft (internal state), move itself to the target computer and notify all

> of its neighbors that it was moving and to where so that communication with the neighbors could be re-established when the move completed.
>
> To begin to address the technical problems posed by moving a part of the ongoing distributed computation I decided to see if I could build a simpler non-distributed program, which I named Creeper. Creeper's job after being started on one computer was to pick itself up and move to another by sending itself across the ARPANET to the other computer. To make its job a little harder, I required Creeper to perform a simple task without error as it moved from computer to computer. Creeper's simple task was to continuously print a text file on a console without missing or repeating any characters. To perform the task without error as it moved from computer to computer, Creeper had to bring the file being printed along with it and to keep track of how much of the file it had already printed before it moved so that when it resumed printing after moving it would resume at the right point.
>
> After getting Creeper to work, I did two things. One was to integrate the techniques it used into McROSS so that parts of a distributed simulation could move around the ARPANET as the simulation was ongoing. The other was to hack Creeper giving it the capability to wander aimlessly (and endessly) through the various DEC PDP-10 computers on the ARPANET, picking its next host at random. Someone, I'm not certain, but I'm pretty sure it was Ray Tomlinson, decided to write as another hack a Creeper killer — I think he called it Reaper — which would seek out and destroy Creeper. Once Ray debugged Reaper it never failed, as I never added any defensive mechanisms to Creeper.
>
> This all seems pretty primitive now, but at the time it was exciting and a helluva lot of fun. Almost everything anyone thought of or tried with the ARPANET had never been done before — pretty exhilarating stuff!!

In May of 1972 Thomas reported on his experimentation with a Multi-tasking Telnet implementation for TENEX, in RFC 339 "MLTNET: A Multi-Telnet Subsystem for TENEX." This capability was one of the first instances of trying to take advantage of the parallelism opportunities that access to multiple machines provided. It allowed a user to multi-task by having and switching attention between multiple, simultaneous Telnet connection to various host destinations,[9] buffering any output accumulating while a user's attention was on another connection. A bit later, Thomas would also report on a protocol for automatically redirecting the end point of a host to host connection from one site to another, in RFC 426, "Reconnection Protocol." This work, an outgrowth of the earlier McROSS experience, was perhaps the first instance of a capability for enabling a computation to be automatically directed to another host (presumably to continue the interaction) without manual intervention.[10] Also, in January of 1973, Bob Bressler and Bob Thomas report on early experimentation with inter-entity (i.e., user-to-user) experiments, in RFC 441 "Inter-Entity Communication — An Experiment." That note outlines some of the earliest thinking about inter-host terminal linking, based on extending concepts which had appeared within timesharing hosts. This sort of terminal linking was used as an early version of what we now know as instant message for direct back and forth messages, and as a useful utility for doing remote debugging or online help.

By 1973, Thomas' investigation of services for networked computing became focused on providing a more general operating system runtime environment useful for distributed computing environments, in the form of a resource sharing executive (RSEXEC). The air traffic application and the experimentation with worm-type behavior, while showing the potential for future innovations, were bypassed in favor of more immediate needs in making the networking capabilities easier to use by more people. That became a driving theme for much of the distributed computing work for decades to come.

One other of the early investigations deserves mention, because it marks the beginning of the fuzzy boundary between the network communication and the higher level forms of communication which serve as the base for distributed computation using the network. In addition, it highlights an area of constant debate and path reversals over the entire lifetime of these Internetwork activities. In parallel with BBN scientists working on defining the IMP capability, a network working group, chaired by Steve Crocker of UCLA, was thinking about software to utilize the emerging capability. The first order of business for this group was the definition of an overlay communication subsystem called the Host-to-Host protocol. This would be what software embedded in the host operating system would use to communicate over the ARPANET to another host on the network. The innovation with the IMP technology was packet switching, whereby information from one host to another consisted of messages, which were broken into packets, each of which might traverse the network separately, to be assembled into a complete message at the destination.[11] Despite this underlying architecture, the first Host-to-Host overlay protocol, called the Network Control Protocol (or NCP) was connection oriented. That is, it had a distinct phase for setting up a direct, point-to-point connection between two specific hosts, a phase for streaming data between the hosts, and a connection shutdown phase, as part of the standard interface to the network. This would be the start of a long and continuing debate in the community over the merits of message-oriented vs. connection-oriented approaches. Over time and for different applications, the technology has flip flopped between these approaches, sometimes taking connections and using them to construct a message passing capability on top, and sometimes taking message based capabilities and using them to facilitate streams of messages between hosts. In any event, establishing the first layer above the IMP message and packet switching as a connection-oriented protocol was not without some controversy. Although many of the applications envisioned may have been stream oriented with long-lived connections between entities, much of the operating system research and development of the time centered on message passing systems [12]. So it was not surprising that opinions differed at the time for host-to-host protocols.

Some early evidence of this comes from Dave Walden. Walden was a participant in the small team of engineers that developed the first packet switch, the ARPANET IMP. The NCP end-to-end connection system[13] puzzled Walden, who observed that the ARPA host-to-host protocol that was under development was a system for communication among *operating systems*. He argued for a system based on communication among processes running on operating systems rather than among operating systems, and sketched the primitive operations for such a system. He further argued that future operating systems should be designed with such network-based interprocess communication in mind and that the ARPANET protocol for communication among operating systems, at the beginning of the modern data communications era, was a step in the wrong direction. This point of view reflected similar thinking that was emerging at the time in the operating system research community.

In 1970, Walden drafted RFC 61[14] suggesting an alternative system without end-to-end connections. Walden quickly revised and republished his ideas as RFC 62.[15] At the time, it was entirely an intellectual exercise, as there was no coding or implementation planned for an alternative Host-to-Host protocol. To further promote his ideas, Walden turned the RFC into a paper that was submitted to the *Communications of the ACM*[16] and also submitted a version of the paper to Norm Abramson's annual conference in Hawaii.[17] Walden reports that the CACM referees were very enthusiastic about his ideas; the paper was reprinted later in a compendium of important networking papers of the time.[18]

For his Master's thesis at MIT,[19] Bob Bressler implemented and investigated Walden's ideas and was able to show full functionality between two PDP-10s with a much smaller implementation than for NCP, and he did some performance investigation. Bressler and Walden became acquainted through Bressler's interest in Walden's idea, and Walden encouraged Bressler to join BBN. Bressler did join BBN, did technical work on several systems related to BBN's ARPANET and packet-switching work, and eventually took over management of the part of Frank Heart's division that Walden had been managing when Walden founded BBN Information Management Corporation and developed InfoMail (Chapter 19).

On May 15, 1972, Bressler, Dan Murphy,[20] and Walden published RFC 333, suggesting a significant experiment with a system like that sketched in RFC 62 and refined by Bressler as part of his MIT thesis. RFC 333 again argued for the power and simplicity of a system based on switching messages rather than the "stream orientation" of the ARPANET NCP. Similar investigation of a network IPC support for message passing was ongoing in the technical community at SUNY Stony Brook, where Rick Schantz was pursuing the idea of "An Operating System for a Network Computer" as part of his PhD thesis work,[21] as well as at UC Irvine with Dave Farber and the DCS system[22] he was developing. Schantz would also soon join BBN to continue working on pursuing the message passing as the basis for distributed computing ideas,[23] and Farber would also become heavily involved with Internet communication activities and evolution.

A short time later the work began on what became TCP and later TCP/IP, and a compromise of sorts between the message passing and connection oriented schools of thought. In 1996, Peter Salus, who had discovered and been excited by RFC 62 while researching his book Casting the Net,[24] convinced Bob Metcalfe to include a copy of Walden's RFC 62 in the republication of Metcalfe's PhD thesis.[25] Metcalfe's thesis, which was an important precursor to his invention of Ethernet, mentions Bressler's and Walden's work relating to RFC 62 and RFC 333 as an alternative to the connection-oriented NCP specified by Carr, Crocker and Cerf. In his Overview, Salus described the RFC 62 ideas as "a 'road not taken,' perhaps to our loss." In addition, subsequent work would provide many variations of the message passing approach, albeit often at higher levels of the protocol stack. Walden reports that he likes to think some of his ideas, which were well known to Vint Cerf and Bob Kahn, influenced TCP/IP in some small way. In any case, RFC 62 was an early effort to think about interprocess communication in today's network context and anticipated the message passing systems of the future.

18.3 Distributed computing infrastructure emerges and evolves: 1973-1995

The evolving infrastructure grew out of ideas developed from a number of sequential activities, each covering many years, and each building on the lessons learned from the earlier work: RSEXEC, a homogenous operating system; National Software Works, a heterogeneous operating system; a number of algorithms, studies, and related experiments; and Cronus, a distributed object computing system.

RSEXEC (homogeneous) distributed operating system: 1973-1976

The early 1970s saw the widespread adoption and use of the BBN developed TENEX virtual memory time shared operating system within the DARPA research community. By that time BBN was already running at least 5 of these systems for its own computing projects, each of which serviced from 20-40 users in time sharing mode, with continued growth likely. In that context, in 1972 Bob Thomas began a project to bring the emerging distributed computing visions to the TENEX world. That project became known as RSEXEC, for Resource Sharing Executive[26] (operating systems at that time were commonly call Executive programs, because they took charge of organizing the running programs on the hardware base, and on TENEX the command interpreter was called by the program name "exec." RSEXEC was in essence the extension of the exec to incorporate resource sharing across TENEX hosts). Its goal was to pool together the resources of the TENEX systems running at BBN and elsewhere throughout the now being connected DARPA research community, and develop the operating system type of support for transparently using resources on any of these systems. Rather than being driven by any particular application, this project addressed the technical challenge of extending the reach of operating system concepts across hosts.

This first generation of our so called "network operating system" provided prototype solutions for distributed computing in a homogeneous systems environment, based on a message-passing paradigm. TENEX had a rather sophisticated (for the time, and likely since) and flexible runtime structure for dynamically creating groups of processes which could cooperate on tasks, as well as access and share file information stored in a rather elaborate hierarchical file system. The innovation of RSEXEC was to provide software extensions to this model so that the creation and accessing of processes and files (and other resources as well, such as printers, tape units, etc.) did not stop at the host boundary. Rather, through RSEXEC extensions to TENEX, it was as though the collection of TENEX systems accessible through the network formed a virtual computing capability, in much the same fashion as the TENEX system did for the resources of one host. Working with Paul Johnson, a programmer in the TENEX group, Bob did what BBN was becoming known for: rapidly developing a working version of the software as an evolving concept demonstration and experimental vehicle, while making the software solid enough to be available for trial usage. By 1973 there was a running prototype of the RSEXEC software on all of the BBN TENEX machines, as well as on other TENEX hosts from sites who wanted to cooperate in the experimentation. Some of the innovations first demonstrated and provided by RSEXEC are described in the next paragraph.

A network file system, made up of the collection of files on the cooperating TENEX hosts, was available to both users at keyboards for use with "exec" commands (e.g., TYPE file, COPY file, Delete file) and for executing programs which could open/read/write/close files from anywhere within the confederation of TENEX systems. RSEXEC supported a distributed file system file structure which permitted the "mounting" of a host directory into a global file system hierarchy. Through the use of daemon servers running on each participating host, commands referencing remote files would be relayed to the appropriate server for executing locally and passing the results back across the network using a message passing paradigm, as shown in Figure 18.1 taken from early reports on our activities. Through this strategy, virtual paging of the file data was extended to work across the network interface for remotely accessed files. Another key innovation introduced by the RSEXEC work was the concept of providing an operating system extension to "trap" system calls from running programs before they are serviced by the local resident operating system.[27] This "JSYS Trap" facility (in TENEX, system calls were issued by the jump-to-system (JSYS) instruction using a code to indicate which service was requested) was a key development in providing an overlay environment for executing processes which served to extend the reach of system calls (where appropriate) across host boundaries. This capability was the start of providing extended virtual machines by reinterpreting machine instructions in an extended context. So when a program executed a system call (for example, to create a new child process) RSEXEC software, interposed between the program and the operating system, would get control to interpret this system call in the context of the collection of cooperating machines. The request could be passed to a daemon program on the appropriate machine for

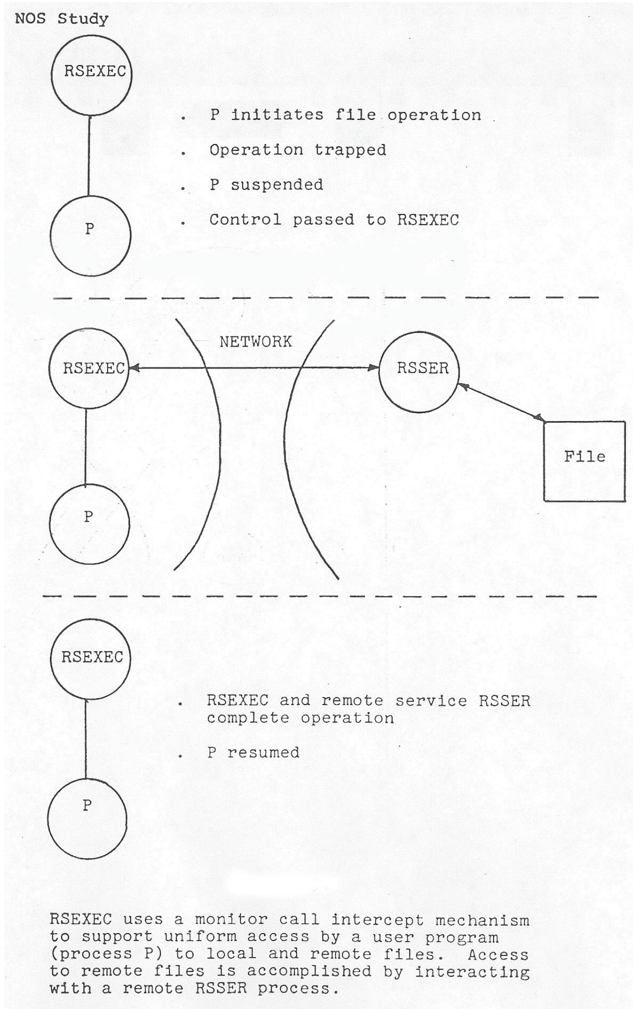


Figure 18.1 Supporting network transparency in RSEXEC.

executing remotely, with the results relayed back to the trapping program for returning results to the originating process, or be allowed to pass to the co-resident operating system for executing locally, whichever was appropriate. These simple functions of interposed system software, remote daemon server processes, and extended sets of collections of system resources from the collection of cooperating hosts, reachable and useable by running programs through the interposed system software, would form the architectural footprint for distributed computing for many years to come.

While this work was going on at BBN, Rick Schantz was finishing up his PhD dissertation at SUNY Stony Brook, also investigating issues in the design and development of network oriented operating systems. These ideas were similar to those being pursued by Thomas with RSEXEC, and in 1973 Schantz came to BBN from Stony Brook. He joined the TENEX group to work with Thomas on nurturing the ideas behind the emerging field of distributed computing, a direct outgrowth of the growing interconnectivity made possible by the complementary ARPANET IMP and NCP protocol work going on at BBN and elsewhere in the ARPA research community. Through 1974, Thomas and Johnson, with some support from Schantz, refined and extended the RSEXEC system such that it was a viable utility running on many of the TENEX hosts throughout the ARPANET (and a few non-TENEX hosts, in a more limited way, as well). We continued to pursue the advanced interprocess communication themes.[28]

By 1975, IMPs were proliferating and TIPs (Terminal Interface Processors)[29] were invented to provide a relatively low cost access point to the network without having to go through a large time-sharing host. The ARPANET was a useable utility (albeit largely by the computer science R&D community) and its use prompted the inevitable concern for controlling the cost and accessibility of the capability. By this time Vint Cerf was at ARPA and responsible for the network, and was soliciting ideas for how to manage and control the use of TIPs to access the ARPANET.[30] The problem was that the TIPs were very limited devices, with no possibility of hosting complex access control and accounting software. Dave Walden floated the idea of using the ubiquitous TENEX systems by adapting the existing RSEXEC services to augment the limited capabilities of the TIP/IMP environment. This idea was accepted and Thomas, Schantz, Walden and Bernie Cosell set about to design and implement a robust, highly reliable, redundant capability for the large TENEX machines to provide an access control (i.e., TIP login) capability, as well as a packet accounting subsystem (i.e., recording and collecting message traffic data to facilitate eventual billing for services provided). As was becoming usual, this new service was to be developed and demonstrated within a few months, and by the end of 1975 there was support for an operational capability. Some of the design and implementation considerations in these new services were reported in RFC 672, R. Schantz, "A Multi-Site Data Collection Facility,"[31] December 1974, and RFC 677, P. Johnson and R. Thomas, "The Maintenance of Duplicate DataBases," January 1975. This was perhaps the first running example within the ARPANET of using the resources of a large host to augment the resources of a small, limited host in providing advanced services beyond the capability of the small host alone. In addition, innovations were introduced in the areas of keeping data bases replicated as a means of providing reliable service even if there were outages, and in the area of load sharing among a large collection of potential servers for collecting the accounting data periodically from the TIPs, and then reconstituting a continuous set of data for billing purposes which reordered and removed redundant copies which may have been collected.[32]

Although this capability was never put into actual service, largely because of the centralized registration (no one at DARPA wanted to be handling the requests to add or delete users), it did represent breakthroughs in terms of how to transparently augment the services of limited capability machines by off-loading functions across the network, and in how to organize collections of servers to provide robust and responsive services under varying load and failure conditions. TIP access control was eventually provided at a later date, using another technology solution (and distributed update controls), but TIP accounting never seemed to surface again, in favor of the unmetered service we have to this day.

By 1976, RSEXEC was operating daily on a wide collection of TENEX systems throughout the ARPANET. In addition to supporting the TIP small host augmentation capabilities just described, it was used extensively by us at BBN to do our computing over the (by now) five separate TENEX systems we were operating and using, and by a number of other colleagues to augment their home computer resources. Although widespread support for the system never materialized, we believe it represents the first use of network operating system concepts operational over the wide area ARPANET communication network.[33]

Bob Thomas recalls the evolution of ideas and purpose of the work in distributed computing research and development:

> The early work (early- to mid 1970s) resulted largely in prototypes that served as feasibility demonstrations and laboratories for identifying and addressing technical problems associated with using a network to support distributed computing.
>
> For example, the McROSS work demonstrated the ability for multiple computers linked by the ARPANET to cooperate on non-trivial computations. It also demonstrated the ability to move such a computation from one computer to another with no impact other than a temporary slow down. The creeper worm was a simple hack that convinced me that "real" computations, such as parts of a simulation could be built with the ability to move from one machine to another. In addition, McROSS showed the ability to use the network to connect a display device (for McROSS an IMLAC graphics station or an Evans & Sutherland display processor in a PDP-10) to a remote computation for the purpose of displaying output from the computation (for McROSS an air-traffic-control type airspace display) in "real time."

The RSEXEC work that followed showed the feasibility of a distributed file system (functionally very much like early versions of Sun NFS) that supported the notion of mounting remote file systems and program access to remotely mounted files - not surprisingly, paging across the 50Kbit ARPANET was pretty slow. It also demonstrated the ability of network server machines to support less capable terminal access machines (TIPS, TACS) initially via 'tip news' and later evolving into the "tacacs" access control system,[34] which is still supported to this day by some router vendors (albeit with a different implementation). In addition, RSEXEC showed the feasibility of inter-host terminal linking and the "terminal advise" feature, early pre-cursors of today's instant messaging and shared workspace applications.

In retrospect our later work (late 1970s through the 1990s), while still having a significant "research"' element, built upon the earlier work but was more focused on developing systems for (usually limited) deployment for "real" users. This work included work in the NSW project and development of the Diamond multi-media e-mail system (and the conferencing and video systems that followed it) and the Cronus distributed computing system.

NSW (heterogeneous) distributed operating system: 1975-1982

While we were busy at BBN developing a distributed operating system for TENEX, ARPA was busy seeking other innovative uses of the emerging network. Steve Crocker, who as a graduate student at UCLA was instrumental in leading the working group that developed the ARPANET Network Control Protocol (NCP, the predecessor of the now ubiquitous IP/TCP suite), was now an ARPA Program Manager and initiated a program around 1975 with ambitious goals called the National Software Works (NSW).[35] The idea was to promote resource sharing by making available suites of software development and programming support tools that were available on various of the computer systems connected to the network, and be able to use them from anywhere in various interoperable combinations from a common repository spread nationwide across these hosts. In what was becoming a preferred ARPA style of doing business in the networked era, a collection of participants with a variety of expertise and ideas in the area were selected to pursue the project. BBN, despite significant capability in the area was not among them. In all likelihood this was at least in part because of ongoing disputes between BBN and ARPA over release and ownership of some of the work products of earlier projects. However, we did initiate comments on the technology that was being developed under the NSW program, as members of the (by now electronically linked) ARPA supported research community. These "debates" centered on the ongoing question of the day relating to the relative merits of message passing vs. procedure calls as the basis for a distributed computing paradigm. RFC 674, "Procedure Call Protocol",[36] and RFC 684, "Commentary on Procedure Calling as a Network Protocol"[37] documented the ideas that were emerging at the time.[38] The research at BBN was message passing (operating system) centric, while the NSW was emerging as procedure call (programming language) centric. So when the NSW program seemed to be floundering in getting off the ground, Schantz went to a meeting held at SRI to discuss what could be done. Shortly after, BBN was invited to join the NSW program after outlining ways in which the emerging RSEXEC experience could be used to jumpstart the NSW program, especially on the TENEX host. Thus NSW became the next opportunity to address the challenges of developing a network operating system, our second generation network operating system, this time to support interchangeably running programs across the various types of hosts connected to the ARPANET, including MULTICS based hosts and IBM OS/360 based hosts, as well as TENEX based hosts. In other words, this would be very heterogeneous from the outset, in contrast to RSEXEC which was largely homogeneous around the TENEX concept. This distinction between homogeneous and heterogeneous assumptions and emphasis is one which still exists and continues to be fervently debated to this day.

Whereas the current BBN research was being carried out by a single group of like minded individuals, the NSW project was being carried out by teams from Massachusetts Computer Associates, SRI International, MIT, UCLA, and now BBN, with very different expertise and perspectives.

NSW was intended to provide managers of programming projects access to a collection of management tools for monitoring and controlling project activities and give programmers uniform access to a wide variety of software production tools, as well as experimental tools being developed as ongoing software engineering research and development. Although many such tools existed, and were available on a variety of different computer systems, they were never applied together in an effective way to support a software implementation project because the capabilities were dispersed among the various kinds of hosts now connected to the network. In essence, NSW was the first exploration of networked resource sharing focused on sharing computer programming utilities across a heterogeneous environment, accessible from anywhere on the network. The main impact of the work, as far as we were concerned, lay in the distributed system infrastructure and services that would be required for such an advanced system, and we still viewed that entity as a network operating system. Working with Bob Millstein, Charley Muntz, Kirk Sattley and Steve Warshall of MCA, Jon Postel, Jim White and Charles Irby from SRI, Doug Wells from MIT and Bob Braden and Neil Ludlam from UCLA, Schantz and Thomas reworked the preliminary concepts to put them into the form of functional elements tied together by network operating system concepts, which we knew could work from the RSEXEC experience.

A useful view of the principal components of the NSW system is as processes that cooperate to provide network wide services. These components included a Front End (FE), which served as the host independent user access point, a so-called Works Manager (WM), which served as a centralized resource allocation and access control module, and a variety of so-called Tool Bearing Hosts (TBH), each of which provided two services, a Foreman (FM) providing the runtime environment for running tools/programs, and a File Package (FP) which was responsible for managing the files stored on the local TBH and managing the transfer of files between the File Packages of the TBHs as needed to run the various programs. Each active user, from anywhere on the network, had a dedicated front-end process which served as his interface to the NSW system. The principal function of the FE was to interpret the NSW command language and make requests of other components as necessary to satisfy the user requests (for resources scattered among the NSW hosts). The WM maintained databases of user password and account information, NSW file system catalogs, and program/tool descriptor information which was needed to find and request the startup of the tool on an appropriate remote host. All requests for access to tools and NSW files went through the WM. When a user requested a service through his Front End, the request was forwarded to a WM process to both find and check access for the appropriate tool. That request was forwarded on to a Foreman process on the selected TBH, which established a runtime environment to the tool which would be connected directly with the FE for interactions with the user, and with the WM for access to files from the global file system. When files were accessed by the running tools, they would be automatically located, transferred and translated, as needed, for the particular tool through interactions among the various File Package processes, as depicted in Figure 18.2.[39]

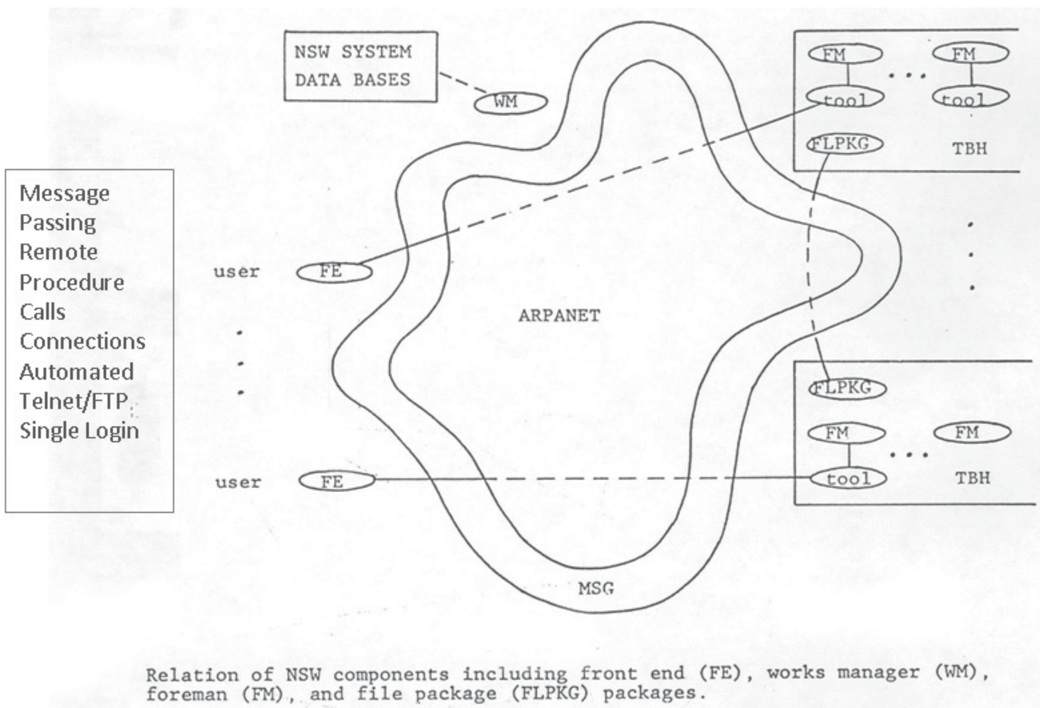


Figure 18.2 Components of the National Software Works programming model.

There were a number of insights, innovations and practical services developed in the course of putting together an infrastructure that could effectively support all of these interactions over the computing capabilities of the time. We briefly highlight a few of these.[40]

In order to make all of these interactions work across the heterogeneous platforms, BBN (along with participation from other NSW participants) undertook the design and development of a major network infrastructure enhancement component which would handle all of the communication needs for these system components. The subsystem developed (called MSG, for message system) was an advanced, heterogeneous interprocess communication capability overlaying the ARPANET communication capabilities.[41] MSG introduced a number of key innovations, most prominently in specifying "type" information about the services provided by an MSG registered communicating process, and by supporting two types of addressing modes, generic and specific. Generic addressing was the means by which a process initiates a transaction with another previously unrelated process. It is used when any process of a given type is acceptable (e.g., a new WM process to service a new session). Using the "type" information of the various registered processes, generic requests can be routed to any server supporting that type of service. Specific addressing is used when the sending process must communicate with a particular process only (e.g., a specific FE where the user resides). When a process issued a receive operation, indicating it was now willing to take a new message, it declared whether it was requesting a specifically or generically addressed message (i.e. whether it was in the middle of a specific transaction, or it was available for "new assignment").

In order to run programs within the distributed NSW environment, many of the ideas first established in the RSEXEC experience were used and extended to the heterogeneous NSW environment. Tools could be either "batch" type tools (this was the predominant mode for tools running on the IBM OS/360 hosts), in which case all of the required setup, including file transfers and transformations, would be done prior to running the tool, or "interactive" type tools (this was the predominant mode for tools running on the TENEX and MULTICS time sharing hosts), in which the tools would be run in an encapsulated environment, dynamically requesting and transforming resources through the intelligence of a "Foreman" (FM) process interposed between the tool and the "Tool Bearing Host" (TBH) system.[42] This sort of encapsulation led to significant activity addressing the transparency issues associated with inserting a new operating context for existing programs (vs. rewriting programs for a new, distributed environment).

Another major technical focus area for the NSW work was in the nature of the interactions among the components. Much of the initial ARPANET-based work on interactions focused on protocols among the cooperating (and often heterogeneous) entities. That was appropriate as the focus was on arms length common interpretation of handshaking and transfer interactions among the completely separate machines. Systems like NSW focused their attention on more of a distributed computation, which had elements that ran on different hosts (e.g., a request to run a service or to access a resource, which involved the interaction of a number of cooperating entities in a standard way, independent of the type of system). To do this in a more predictable manner, NSW (and systems like it) began to focus on specifying and standardizing interfaces on these hosts as well as protocols between the components. Thus the common agreement was not only what messages to send but also in the semantics of the services available to the program making the request (e.g., every TBH FM would have a "run program" interface; every WM would have a "login" interface for a single login to obtain services anywhere). In a homogeneous environment (such as TENEX RSEXEC) one could merely use the local OS version as the standard for whatever primitive operation/system call was required among the participating hosts. In a heterogeneous environment this was no longer possible, and emphasis on common interfaces emerged. In addition, work was starting on taking a programming language approach to interactions among cooperating components. Work of this kind was ongoing at SRI International through Jim White, Jon Postel and Charles Irby and others, who were participants in the NSW program. Based on work that they initiated, the messages (requests and replies) that were being passed through the MSG IPC capability were actually encoded using an RPC like convention. This was likely one of the first operational uses of a procedure call orientation to "gluing" components together, and certainly the first to marry the RPC notion with an advanced IPC message passing paradigm. This combination would serve to support our advancing ideas on distributed computing for decades to come.

By 1976 Bill Carlson had taken over from Crocker as the ARPA NSW Program Manager, and he initiated a focus on moving toward achieving a capability that could be used and evaluated. An operational NSW system was completed by 1977,[43] and went through a number of revisions to improve its usability, its reliability, and most notably, its performance (the operating platform was, after all, still geographically dispersed throughout the country concentrated on the east coast, medium sized time sharing hosts with NSW system software running as application code outside of the native operating system, over ARPANET connections with a maximum capacity of 50kbs, resulting in significant costs for interprocess interactions which formed the basis for the system). The co-sponsor of that work became the Air Force Rome Air Development Center (RADC, currently known as the Air Force Research Laboratory's Rome Research Site [or Information Directorate]; they would later be important in picking up the ball from DARPA and sponsoring their own Distributed Computing program, with much help from BBN. RADC, predominantly through Tom Lawrence and Dick Metzger, was ARPA's original DoD project manager in the NSW work and our contracting agency. In supporting the refinement of the original ideas into an operational capability which would be sufficient to undergo test and evaluation use trials by Air Force users, RADC became the focal point for managing the activity .[44] One of the key BBN programmers in this transition activity was a young, very productive system developer named Steve Toner, who had recently joined the Distributed Computing group after graduating from MIT. Steve had worked part-time for BBN as a night time system administrator for our TENEX systems while in college, through an ongoing arrangement we had with a fraternity at MIT.[45] Steve did his system hacking at night, and was an obvious talent. We became familiar with him, and he became familiar with us, and this was one of a number of examples of hiring as students, and eventually fulltime, a stream of bright, energetic and technically curious individuals, often from local universities, particularly MIT.

By 1978, an effort had been completed to "productize" the system (not in the sense of having a product to sell, but rather in the sense of operating a utility-like capability which could be operated and maintained during the course of an extended period of training and use by Air Force users). Three centers of the Air Force Logistics Command (AFLC, in Sacramento, Ca., Oklahoma City, Ok., and Warner Robbins, Ga.) were selected as the initial user community for test and evaluation of a system for resource sharing and remote tool access within and across the centers. Appropriate tools were selected and inserted based on the needs of the evaluation participants. Training courses and user documentation were developed, and an operational staff was assembled and put in charge of managing the system and its trial use. To our knowledge, this represents one of the first large scale, wide area, heterogeneous network operating system based application trials ever conducted on the ARPANET. These trials were successful in introducing the technologies to a wider community, and in evaluating the particular instances of them. Remote access could clearly be supported and system boundaries could be sufficiently blurred to make them reasonably transparent to the user. However, the attraction of the particular tools and services was limited for a domain focused group such as AFLC, and the integrated systems capability was far from utility-like, and very difficult and expensive to operate and maintain. Much was learned about supporting these types of systems in real operating environments, and the trials ended by about 1981.

An interesting sidelight to this activity was the fact that in order to participate in the experiments, an ARPANET presence needed to be established at each site. At this time, ARPA and DoD were very interested in supporting the spread of the new communications technologies into DoD installations. While each individual center seemed interested in participating, there was less enthusiasm for the implications of resource sharing between the centers. However, the cost of participating was accepting the installation of an IMP connection for the bases. While the experiments with higher level resource sharing lasted less than a year, the interconnection of the sites (for other purposes) likely has had a much more lasting effect on the organizations and the growth of the Internet.

One of the obvious shortcomings of having a centralized resource allocation component such as the WM was its effect on system performance (single host overload) and on system reliability (no WM host, no service). The need to have a common database to drive the WM functionality limited the initial vision to a single host, multiple process implementation. The limitations of such a design were obvious, but took a back seat to issues of developing a prototype capability. By 1977, there was already a significant activity mounted to design the WM as a multi-host capability, and with it, tackling the difficult problem of maintaining replicated data bases. We had already begun to investigate the replicated data consistency issue[46] in the context of other such needs (e.g., TIP access control), and in the course of doing so developed the ideas underlying the majority consensus approach to replication management, whereby replica owners vote on transaction operations, with a quorum of those eligible to vote required to commit to a change. As a result, we were very familiar with the difficulty and technical risk of such an undertaking, and proposed an alternate interim-reliability plan instead. That plan focused more on recovering from temporary failures and outages (still very common in the relatively early days of ARPANET and timesharing) by ensuring that users lost no completed work products if the system suffered a failure of any of its critical parts. That interim plan, focusing on checkpointing and recovery instead of replicated consistency, was adopted and implemented by mid-1977. Although we never achieved an operational multi-host WM capability, the issues raised in handling fault tolerance and load balancing, and the work on the various ways in which to keep replicated data synchronized, paved the way for many and repeated returns to this problem over the course of the next 20 years, especially influencing the directions taken in our third generation systems. These issues are still the subject of much controversy and continued research and development as to how best to manage the tradeoffs between performance, accessibility and resilience.

Another factor in the fading away of the NSW system was the rapid evolution and change that the computing landscape itself was undergoing. During its lifetime, the NSW project needed to deal with the changeover of network substrate from the NCP protocol to the newly emerging IP/TCP suite, as well as with the migration of TENEX to Digital Equipment Corporation as their TOPS20 operating system. Each such change, and others like them, required significant modification to the network operating system layer which integrated the pieces together into a single system. In addition, new classes of computer systems (workstations) were emerging, along with other forms of operating system (Unix) which also began to gain acceptance. The NSW code base was mostly assembly language for the large-scale time-shared OSs of the day. The cost of maintaining this system in an era of rapid change signaled that the NSW project was coming to a close. Though these technical investigations were very successful, the impact of applying them economically on the prevailing computer technical infrastructure was less so.

Algorithms, mechanisms, studies, designs, experimentation support: 1976-1980

By 1976 Harry Forsdick had joined the Distributed Computing group out of MIT. Harry had more of a computer language orientation, and would be instrumental in leading us toward some of these viewpoints on computing, initially through activities with using optimized BCPL (an early high-level language for systems work[47]) as an approach to improved performance, and later with Pascal-based language systems. But his real impact would be much later, focusing on advanced Internet applications. For now, with added resources our activities were expanding beyond building experimental network operating systems.

Although we (and the DARPA community in general) were firm believers in the experimental approach to Computer Science research and development, we also began to try to formalize the problems we encountered and the solutions we envisioned in studies, reports and published papers.[48] These activities were often spinoffs from our experimental code development activities.

Our experimental systems work was trying to prove the value of multiple systems working closely together facilitated by a common, advanced infrastructure, and two areas of immediate concern were load sharing and tolerating failures of individual systems. Our RSEXEC work had explored the potential of having multiple hosts each store copies of files to improve their accessibility. One problem this raised was in keeping the content of these files synchronized when they changed. This same problem emerged in the NSW work in the form of support for common services which required common databases to operate correctly. From early experiments in 1975,[49] through formal publication by 1978,[50] we had been developing ideas about how to approach the problem of maintaining replicated copies on a network. Bob Thomas, in a burst of concentrated theoretical work, introduced and formalized the ideas behind majority consensus and quorum voting approaches to maintaining duplicate data. In contrast to much of our previous effort, which was focused on building a particular capability, this work became more abstract, organizing the problem and its solution space in general. It was, however, based on the earlier practical experiences of dealing with many copies of files and with many transient failures which were common to ARPANET computing at the time. This seminal work broke with the traditional approach of having a primary copy and a backup, to one of peers voting on the possibly conflicting updates they might each initiate in parallel. By varying the number of participants in the peer group, different tradeoffs might be made between availability, failure tolerance, and the overhead of making and distributing changes. In reality, this work had a more profound influence on the emerging database community (of which we were never really part) under the banner of distributed transactions than it did on the operating system community (of which were a part) under the banner of distributed file systems. But it wouldn't be until the mid-1980s under another generation of distributed system infrastructure that we would develop a usable implementation of these early concepts for flexibly maintaining duplicate copies of data.

Between 1976 and 1980, the team of Forsdick, Schantz and Thomas would be responsible for two major studies performed for RADC to try to develop a more formal footing for the study and evolution of distributed computing infrastructures. The first of these was completed in 1978.[51] This project tried to organize the space of goals, concepts and implementation approaches to developing what we had termed "network operating systems." These ranging from semi-automated use of the now common Telnet and FTP approaches for directly using remote hosts and accessing remote files (which to first approximation outlined the approach ultimately behind the Web and Web browsers, albeit with much improved user interface and graphics); to the more transparent and integrated single operating system vision which was pursued under RSEXEC and NSW; to the various approaches of how to achieve this integration, given the complexities of the network computing on the ARPANET of the 1970s. Of course, our DNA (i.e. the innate nature of the typical BBNer) always favored the most challenging technical direction. One of the key drivers of this study was the experience we had with poorly performing systems operating over wide area environments, which led to an emphasis and speculative paper designs that accommodated more localized operating assumptions. This extended "thinking and planning only" work, although not typical for us, became a nice complement to our experimental computer science orientation and was instrumental in organizing the directions taken later in NSW system performance enhancements as well as helping in formalizing design concepts which would appear in our next generation system.

For the second major study, completed around 1980, we had abandoned the terminology of network operating systems in favor of "distributed operating systems." In part this was because we needed to distinguish one set of activities from the other. But also in part because the computing environment was really undergoing changes. Indeed, the computing landscape at the time was shifting from one of a few large computer systems to many smaller, cheaper systems enabled by innovations in chip fabrication; and from a few large networks to many smaller, locally managed networks enabled by technologies such as Ethernet. Moreover, the transformation from a programming environment dominated by custom assembly languages to a variety of high-level languages (e.g., C, Lisp, Pascal), enhanced the conception and execution of more complex applications. It was becoming clear that the proliferation of new technologies would depose many of the older computing hierarchies. It was clear, too, that the distributed computing focus would have to change to keep pace with the size, scope and variety of the computer technology base.

Even with the early study our perspective had changed from one of building "a time sharing system" across multiple platforms, to one of integrating the interoperation of multiple machines across a distributed environment. The emphasis also shifted from an operating system perspective on processes, files and devices, to a focus on approaches to global system wide resource management and reliability. With the second study project, Distributed Operating System (DOS) Design Study,[52] the formal and paper design work became much more sophisticated, focusing on economics of computing, abstract machines, programming aspects, and general object systems, along with more sophisticated approaches and algorithms for handling physical distribution, relying on timestamps and sophisticated coordination strategies in support of an advanced distributed system infrastructure across the cooperating platforms. Always cognizant of the need for high performance across the systems based on our earlier experiences, by 1980 our paper designs were already dealing with resource management across the various sizes, shapes, and locations of the resources relative to each other. Most of this work was at this stage relegated to simulation and modeling, and a significant part of that was attributable to a new staff addition, Bill MacGregor, fresh from finishing a PhD degree at the University of Texas concerned, in part, with modeling distributed resource management. We had become familiar with Bill and the activities at Texas earlier from working with them on modeling the NSW system as a means of planning performance enhancements to that working system, and he was another instance of adding an additional fresh perspective to our growing focus on distributed computing at BBN. In particular, Bill was a strong advocate for focusing on the emerging work mostly coming out of the programming language community on using objects as a basis for programming. The paper design activities in the DOS Design report laid out the framework of a distributed object approach to distributed computing, a major break with the past. In addition, it focused on the role of the emerging personal computer revolution, and the concept of single role computers, based on much higher level abstract machines. Both of these issues, which we began to explore in the context of this study effort, would have profound influence on what lay ahead.

Before making that conceptual transition, another BBN distributed computing support activity of note occurred during the late 1970s, this time involving the newly networked Unix hosts. The use of TENEX within the ARPA research community was diminishing, slowly being replaced by UNIX, which was appearing on the vastly cheaper PDP/11 family of computers, as the standard development platform. The Computer Systems Division (Div 6) had taken the initiative in promoting UNIX and making it network enabled, as the Information Science Division (Div 4) was still supporting TENEX. Al Nemeth remembers:

> BBN took on a contract (I think starting in 1977 and lasting 3 years) to configure, install and operate large PDP 11/70 UNIX systems at several Navy sites. I remember the Naval Postgraduate School in Monterey, a site in Hawaii, and several more. We did this work, including staffing operators at some of the sites. These systems were in classified environments, and we created a classified facility in Cambridge for the work, with its own PDP-11/70 configuration and encrypted communication to the remote locations. The technically interesting piece of this was our efforts to put in place extensions to the UNIX environment to permit full control of these systems from Cambridge. That included remote monitoring, remote debugging, and remote reboot (without losing control of the management connection to the system), all over ARPANET connections — this was done using NCP communications (as it predated TCP adoption). So, we mimicked some of the Network Operations Center ideas at the host support level for a set of distributed nodes, insisted on standardized configurations, and worked out mechanics for the various tasks. At the time we did this, it was ahead of most of the other efforts that we knew about, and certainly ahead of those in the UNIX community. As a process, it didn't work entirely satisfactorily, requiring us to ultimately place someone on-site in Hawaii - this was handled as a rotating assignment within the group, at first viewed positively, but eventually this type of remote support operation faded as an area of focus for us.

Distributed systems infrastructure and advanced R&D changes gears

By 1981, the seeds were firmly planted to move off in the new directions which had been established by an internal R&D project to build a personal computer which we named Jericho (see Chapter 21), as well as the design studies we had just finished undertaking for RADC. Networked workstations, personal computers, high-level-language-based computing, and distributed objects were going to be rolled in with our growing expertise on developing distributed systems infrastructure.

Harry Forsdick recalls,

> Recognizing the significant changes that were in the wind throughout the computer science technical community and the DARPA technical community in particular, BBN set in motion activities which would enable us to jump in. In 1980, two groups in the Information Sciences Division proposed an internal research and development project to build a "Personal Workstation" which we called Jericho. The original purpose of Jericho was to position BBN as a contender in the bidding for R&D contracts from the government. Although many people complained that we didn't think through a business case where we could commercialize this machine, we did just what we planned to do: build a machine on which we could do research in "Personal Computing" a new term in the early 1980s.

Another internal factor at work was the desire of BBN management to derive more value from the strong team that was focusing on these distributed computing issues. So Forsdick took the route of pursuing a new DARPA program for personal workstations based on the Jericho, while Schantz tried to capitalize on interest the Air Force was showing in supporting the design and implementation of a new distributed systems substrate, carrying forward the lessons learned from the NSW experience, which was winding down. Thomas was active and very influential in both. As it turned out, both of these initiatives in establishing support for new directions were successful approximately at the same time, and in 1981 we were faced with the prospect of starting two major new projects both of which were intending to break new ground and which on the surface had a number of issues in common. (The road would fork again later, as Forsdick pursued Internet applications which became a focus of the DARPA work and Schantz continued with the long running Air Force/Navy distributed systems infrastructure activity, while Thomas branched out to work on support for parallel processing infrastructure and eventually high speed routers within other parts of BBN).

Cronus: Distributed object computing environment: 1981-1995

Although the Air Force was interested in continuing to pursue the distributed computing R&D agenda and BBN had the ideas and interest in performing the research, it was far from simple to get this activity going. RADC did not operate in the same manner as DARPA, and BBN didn't look or behave like the typical defense contractor organization that did most of the system engineering for them. Putting this activity in place was quite a learning process, in both directions, that included three rounds of proposal re-bid iterations and went on for many months.[53] Not only did we have to learn about costing multi-year, multi-phase activities and creative ways to purchase (while sometimes building behind the scenes!) the equipment needed to carry out the research, we had a number of go rounds on the desired approach. Our initial proposal suggested a homogeneous system (think Java) because it would be much simpler to build and operate, and avoided some of the thorny problems we encountered in NSW. The Air Force was adamant about wanting to operate in a heterogeneous environment (think CORBA), and so we changed our approach. Finally in July of 1981, a 3 year contract was let to BBN for "DOS Design and Implementation." The idea was to "define, design, implement and test an Advanced Development Model for a distributed operating system"[54] which could be used to conduct user trials, similar to what NSW had done with prior technology, but preplanned this time. To us that meant we were going to design and implement our third generation distributed system infrastructure, heterogeneous all the way, building on the lessons learned and new ideas spawned from the previous experiences and the design study exercises recently completed. To us that meant building a new type of system, one based around the concept of distributed object computing, and intended for a rapidly changing environment. That system would be named Cronus, after the Greek god who was "lord of Chaos and Ether" (competing local area network technologies of the day). Our intent this time would be to try to ensure these ideas got out of the lab and into transition and common use. While this would happen to some degree, it was a long process with many twists, until the activity was finally decommissioned in about 1995, with parts of it sold to a company (Visigenic Software) pursuing (by that time) industry standard approaches to distributed object computing.

What eventually became known as the Cronus Distributed Computing Environment[55] was a major milestone in the history of distributed computing infrastructure. To our knowledge, it was the first operational implementation of heterogeneous distributed object computing. Begun in 1981 and running through 1995 when the last of the government funding ended and parts were transitioned to a commercial ORB vendor, Cronus became a series of interrelated technology R&D investigations, prototype development projects, application development activities using its advanced capabilities, and test, evaluation and transition activities to ensure the results would be widely available and could easily evolve along with changes in the computing landscape of the day. It was influenced by the earlier National Software Works project, which sought to effectively support distributed heterogeneous software development across the ARPANET. Cronus added object-orientation, equal emphasis on combining local-area networks and wide area connectivity, and the IDL compiler concept. Over its lifetime, more than 50 BBN people[56] contributed over 75 person-years of effort to the design, implementation and transition activities under the Cronus umbrella. By the time it had run its course, Cronus was operational on hundreds of hosts at in excess of 25 sites, served as the basis for numerous operational, advanced concept distributed applications, and provided system training on distributed object computing to hundreds of engineers through week long Cronus workshop courses, run on more than 20 occasions. In 1999, the Cronus system would be honored with a Smithsonian Institution award for technical innovation and included in the Smithsonian's Permanent Research Collection on Information Technology, chronicling the history of computing.[57]

Although the main point of Cronus was our belief that an integrated system infrastructure approach within a coherent system programming model, organized around a distinct middleware layer supporting the collection of capabilities required to build, operate and maintain distributed applications, was the wave of the future, under the cover there were a variety of firsts and technical innovations and areas of taking activities from good idea to production quality implementation which enabled Cronus to continue to be an advanced platform for so long, while remaining astonishingly faithful to its original concepts and architecture. The Cronus system helped carry us forward from the age of software infrastructures for large time sharing systems to the age of middleware centric infrastructures for interconnected, language based development environments running on personal workstations. It transitioned the technology base away from the world of files and processes, monolithic operating systems, and clients and servers, into the world of objects and invocations, small kernels and extensible system services, and language oriented program development. By the early 1990s the transition was well underway to standards based versions of the equivalent capabilities (e.g., CORBA) as the basis for further evolution, and it was time to move to the next stop on the train.

The following paragraphs, taken from the original Cronus system concept report of 1981 summarize what we were after at the time:[58]

> A distributed operating system manages the resources of a collection of connected computers and defines functions and interfaces available to application programs on system hosts. Cronus provides functions and interfaces similar to those found in any modern, interactive operating system. Cronus functions, however, are not limited in scope to a single host. Both the invocation of a function and its effects may cross host boundaries. The distributed functions which Cronus supports are:
>
> - generalized object management
> - process and user session management
> - interprocess communication
> - a distributed file system
> - global name management
> - input/output processing
> - authentication and access control
> - system access
> - user interface
> - system control and monitoring
>
> The primary design goal for Cronus is to provide a uniformity and coherence to its system functions throughout the cluster. Host-independent, uniform access to Cronus objects and services forms the cornerstone for resource sharing that crosses host boundaries. There are two major aspects to the Cronus design: structural and functional. The structural design is concerned with the common framework in

> which Cronus entities operate. This framework makes Cronus a system rather than simply a collection of functions. The functional design defines the specific services within this system framework, and is the major focus for system decomposition ...

All of this was to occur over a heterogeneous platform base: heterogeneous host systems, heterogeneous operating systems, heterogeneous programming languages and heterogeneous communication capabilities. From the outset, there was a healthy tension between function (coming from the operating system world) and structure (coming more from the programming system world). This was indicative of the significant changes which were happening in experimental computer science. We started out as a functional mindset, but soon recognized that the structural aspects would be the concepts that held the system together over the changing computational environment, and would sustain its longevity. Figure 18.3 depicts the relations of the basic parts and concepts for Cronus, from early presentation material.

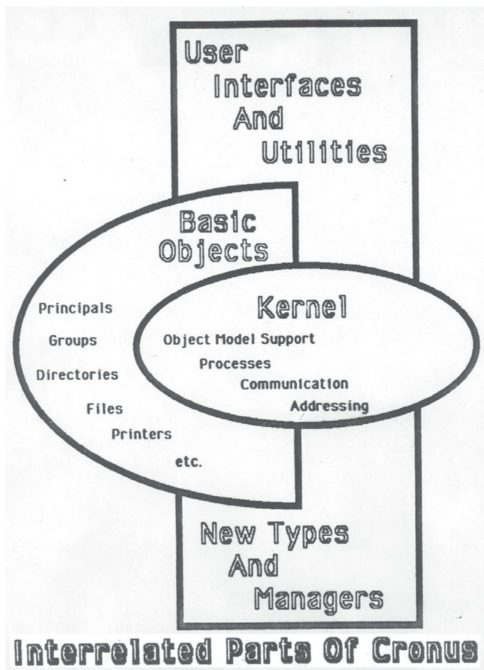


Figure 18.3 A common organizing structure for the Cronus distributed computing model.

Innovations.[59] Throughout its lifetime there were numerous innovations, significant advances and advanced attributes made available operationally as part of the evolving Cronus system design and implementation. These were typically a first operational implementation of these concepts for the Internet environment.

1. *Unified Object Model.* In Cronus, every object was an instance of a type (class) and under the control of a manager (server). This was used to relate the programming system conventions to the client/server model which predominated at the time, and extend it to user defined servers (type managers). Types defined operations, self-defining canonical data types (transmitted in operations and used for persistent state), errors (exceptions), and access control rights. Cronus supported single inheritance; all objects were descended from type Object, which defined standard operations for lifecycle functions, object location, security, self-description, and

management. This was the way in which we introduced the commonality and common implementation across all of the hosts and services, whether system service or user defined service. A given manager often managed several related types as a means of taking advantage of co-location and localized sharing. For each type, Cronus distinguished between generic (a direct descendant of NSW generic addressing) objects and regular instances. Although initially motivated primarily to implement the Create (new instances) operation in a consistent object-oriented way, generic objects and operations were also used as a means of communicating with the underlying manager process ("system") within a host independent common model. Cronus objects were identified by 96 bit global unique identifiers (UIDs) consisting of the IP address of the creating host, a 16 bit object type, and two 24 bit counters. The higher-order counter is persistently maintained and incremented each time the creating host is booted to avoid duplication. In an interesting design decision when deciding how large a unique identifier space we would need, and after experiencing a major software hiccup when the TENEX date counters (number of seconds since some start date) overflowed (similar to but of a lesser magnitude to the year 2K problem), we calculated that with a 96 bit field and some very liberal assumptions on host restarting frequency, a repeated unique number would not be a problem for 100 years(!). Although we never had to worry about repeated unique numbers, the wide addressing used for 96 bit UIDs was a performance issue on available hardware.

2. *Dynamic object location.* By default, Cronus objects could be freely migrated and/or replicated between hosts. The Cronus kernel and Configuration Manager implemented a dynamic object location mechanism. Cronus also supported primal objects (e.g., processes) that were bound forever to a specific host.

3. *Manager Development Tools.* Cronus was one of the first (and certainly the first to reach operational maturity and daily use) distributed systems to employ an interface definition language (IDL) and automated stub generation capability. Our goal was to make the initial construction of a server as easy as possible.

4. *Automated Replication.* The Cronus manager development tools provided for the specification of policies for automated symmetric replication of servers. Cronus replication used a combination of version vectors and voting that could be tuned to favor either consistency or availability. To accommodate outages, each manager engaged in a reconciliation dialogue with its peers when it first came up and at regular intervals thereafter. We believe that this was the first time such flexible replication management policies were made available through higher level specification and tools to automate the process.

5. *Multi-cluster support.* This allowed controlled sharing of servers among clusters which generally corresponded to autonomous administrative domains. Services were explicitly imported and exported between clusters. The use of clusters increased the scalability of the Cronus object location mechanisms by bounding the search space and providing for explicit inclusion outside the cluster.

6. *Protocol-driven user interfaces.* Cronus provided a tool "tropic," a generic client, which obtained interface definitions dynamically from a server, allowing it to invoke any operation. This was an early form of a dynamic invocation capability later provided as part of standard object specifications (such as those promoted by the Object Management Group). These interfaces were used extensively as server debugging tools.

One interesting aspect of this new DOS Design and Implementation activity was that at the outset it was intended to be a real collaboration between the two (often competing) BBN computer divisions, the one that built the ARPANET (Computer Systems) and the one that built TENEX and application oriented software (Information Sciences). This seemed very important, since 1981 was a time of considerable change, uncertainty and opportunity in both the computer systems area (workstations and personal computers emerge) and the computer network area (local area networks emerge). Combining these two areas of expertise seemed ideal. This attempt at close collaboration among and between the divisions' very distinct points of view for example on where the technology should be headed and how to get there, and whether we were an R&D organization or a commercial product incubator and even if we were the latter, which products, was both very effective (early) and difficult to manage (it didn't last very long).

In addition to developing a System Concept,[60] System Architecture and System Design,[61] led by Schantz, Thomas and MacGregor, another early task was establishing a testbed for the development activities. The initial testbed consisted of mixes of various types of platforms to stress the heterogeneity theme and represent a blend of existing and cutting edge platforms, operating systems and languages used. Included were:

- COTS products (PDP 11/70 Unix, VAX VMS)
- BBN's C/70 high-level language oriented entry in the mini-computer market (which included two interesting and sometimes useful features: probably the most viable Unix IP/TCP implementation at the time, and a 20 (!) bit word size, certainly heterogeneous from that point of view)
- Jericho workstations which were being used for Diamond, a companion project (Jerichos were later replaced by Sun workstations)
- small, single dedicated purpose host machines we termed Generic Computing Elements (GCEs, built using a multi-bus based 68000 microprocessor board level products that were driving an emerging market) for which we would develop custom, lightweight system software for one specific function (e.g. network access, file server, and so forth)
- communication gateways (later called routers) to connect the testbed to other such testbeds and the rest of the Internet.

We were one of the early adopters of the Ethernet standard as the LAN technology,[62] and the first at BBN, (after a contentious investigation of alternatives, including rival token passing technology which one of the selection study authors, Ken Pogran had helped develop while at MIT; a product from Ungermann-Bass, one of who's earliest employees came from BBN (John Davidson), as well as a fiber based LAN concept that Jerry Burchfiel et al. at BBN were working on, FiberNet). As history has vindicated, the choice of going with the Ethernet standard was a good one, although getting all of the systems connected was far from simple at that time. Steve Toner was again a key developer getting both new hardware and new software for the many machines to work together. This activity also included developing a custom Ethernet hardware and software module for the C/70 family of computers (the so called MIENI board). As the popularity of the Ethernet began to grow (and new systems, such as PDP 11 family systems, and the Lisp Machine started to be easily configured as Ethernet equipped), what started out as a small testbed network wound up being BBN's campus network, interconnecting multi-access hosts and workstations throughout the BBN Cambridge office, until a corporate network was put in place a number of years later.

Another side effect innovation of the testbed activity was the introduction of the Virtual Local Network (VLN) concept, developed by Bill MacGregor. This was in essence a software layer to isolate the host software from the physical network issues. It tried to isolate many of the issues associated with marrying Internet-style IP based message communication with the underlying Ethernet capability, from the higher level IPC capability needed to drive the Cronus design. In defining the VLN concept, we believe the project was the first to propose and specify what has now become standardized as the Address Resolution Protocol (ARP) to learn about and map between Internet addresses and local host addresses without a preconfigured table (see RFC 824[63,64]).

One of our design goals, motivated by our earlier experience with distributed operating systems, was to be very dynamic and avoid, wherever possible, static entries for configuration or binding, that invariably get out of date and cause the system to break or operate too rigidly in more flexible environments. The address resolution sorts of issues just discussed were part of a more general attempt to support a general approach to dynamic binding of parts of the elements of the distributed computation. To support this type of dynamic binding, there was a dynamic service lookup procedure, again emphasizing no central tables, whereby a host would issue a request for a particular type of service. To facilitate this type of operation, we utilized a broadcast/multicast capability which the Ethernet (and LANs in general) made available, as a means of efficiently contacting collections of hosts searching for a match. This type of dynamic lookup was likely the first of its kind to utilize the emerging capabilities of the LAN technology integrated into a high level dynamic binding facility for communicating entities. To extend this capability efficiently into the wide area, we designed and built what we called gateway repeaters, whose function it was to listen for and repeat the broadcast/multicast from one local cluster to another local cluster.[65] This early use of extended broadcast predates the extensive work done since on a generalized, transport level Internet broadcast capability.

Portability and Heterogeneity. Heterogeneity was a requirement and portability was always a major objective. Cronus development started in C on the BBN C/70 (a PDP 11 class minicomputer with 20 bit words) running Unix Version 7, a VAX 11/750 running VAX/VMS, and a Motorola 68000 “generic computing element” running a minimal executive. Cronus was eventually ported to nearly all current Unix workstations, and to a variety of high performance computing platforms including the Cray, Convex, Encore, Sequent, Alliant, Stardent, and IBM SP/2. In supporting languages other than C, the Cronus approach was to develop native implementations (reimplementing the underlying mechanisms and protocols) rather than language bindings (reusing the C implementation). Although it required more effort, this generally provided better integration with language-specific features such as multitasking, exception handling, and debugging tools. The Common Lisp implementation of Cronus was originally developed for the Symbolics Lisp Machine. It was later ported to the Texas Instruments Explorer Lisp Machine, and to the Lucid Common Lisp and Franz Allegro Common Lisp running on Unix platforms. It used the C implementations of the Cronus system services. The Unix implementations also used the C implementation of the Cronus kernel. The Ada implementation of Cronus was originally developed for VAX Ada, and later ported to the Rational R/1000, and Telesoft Ada. One of the challenging issues we faced in introducing this language heterogeneity was how to access existing code bases that weren’t specifically developed to use the integrating medium of the Cronus object model. The technique we developed was to “wrap” these code bases (e.g., a FORTRAN model) in a translating layer which transformed the object invocation and argument passing semantics into the proper sequences for running the existing code.

These techniques later became industry norms for interfacing existing, non-conforming software whenever new technologies are introduced around them.

By 1982 we had a prototype of the system and its main functional elements working for C on the Unix boxes. Key to achieving this milestone was Girome Bono, a very bright, chess master and Harvard dropout,[66] who built the first Cronus kernel on the C/70. He would spend the next few years reimplementing and refining this code so that it became rock solid and very efficient. Girome was an excellent programmer, even without formal training. This was true of a number of BBN's excellent developers.

In 1983 Mike Dean joined the project fresh out of Stanford and the University of Rochester with lots of computer experience and a can do attitude toward even the most challenging activities. Mike had a hand in many different aspects of the Cronus system, most notably in the manager development tools, in integrating the various different languages, and multi-processor integration. When Mike later moved to the West Coast, he led many of the Cronus based Command and Control applications out of the BBN San Diego office. Mike Dean recalls how he got involved with the project and how he became instrumental in shaping its evolution:

> I had worked with distributed computing at Xerox PARC and Rochester. I interviewed at BBN as a courtesy to one of my professors shortly before Christmas, but really intended to take a job with Multics development (by then part of Honeywell). Over the holidays I read the Smalltalk 80 book, became intrigued with the idea of applying such object-orientation to a distributed operating system, and decided to go to BBN instead.
>
> I arrived at a time when the Cronus architecture and communication infrastructure were in place, but a huge opportunity remained to shape the higher-level software. My goal was to make it very easy for programmers to develop new Cronus servers, and exercised the resulting tools by constructing the initial implementations of a number of system and application services that were among the first to be fielded.

During the early stages of development, we were still touting our work as a distributed operating system. This caused confusion in some people's minds because it wasn't a direct extension of what they understood to be an operating system, i.e., Unix. (This was way before Microsoft helped completely obscure what was the operating system, and what were supporting services, e.g., user interfaces and auxiliary functionality such as web browsers.) In fact, one indirect objective of the Cronus work was to push the programming interface up a number of notches, completely encapsulating the OS, and thereby making it easier to remove it altogether. (A significant number of people, including many developers of Cronus were disappointed with the directions Unix was taking the OS community, after having the more elegant experience of using TENEX. So masking Unix, even if it were underneath driving the hardware, became a desirable trait.) DARPA, on the other hand was already heavily supporting what they thought would be the next generation OS, and that would also be the next generation of Unix. That project, Mach at CMU, had some objectives in common with the distribution goals of Cronus (e.g., kernelized implementation, support for remote services, . . .) but differed significantly in approach, as well as in providing a Unix veneer over the new work to be compatible vs. trying to hide the Unix interface. By 1983 DARPA also wanted to consolidate the technical community around one operating system approach, and in fact called a meeting/workshop trying to resolve this issue. Since what we were doing with Cronus was significantly different from what Mach was doing in providing a new underpinning for Unix, and since the common use of the term operating system in what we were both doing caused apparent confusion, we changed the name of what we were doing to developing a Distributed Computing Environment (DCE). Henceforth the name would be the Cronus Distributed Computing Environment, which in retrospect is probably more accurate, given people's understanding of what an operating system was and still is. The DCE concept became a layering on top of traditional operating systems, or what eventually became more commonly known as middleware, which is what it actually was for quite some time now in our activities. From the beginning, we felt that building the multi-host coordinating software outside of the OS kernel was the most effective approach (assuming the performance limitations could be solved). Now that approach was formalized, and we no longer called what we were doing a distributed operating system. That term had come to mean things like a Novell network for running file servers and shared devices. Figure 18.4 shows an early artifact we used in making distinctions between the different types of system software of the period.

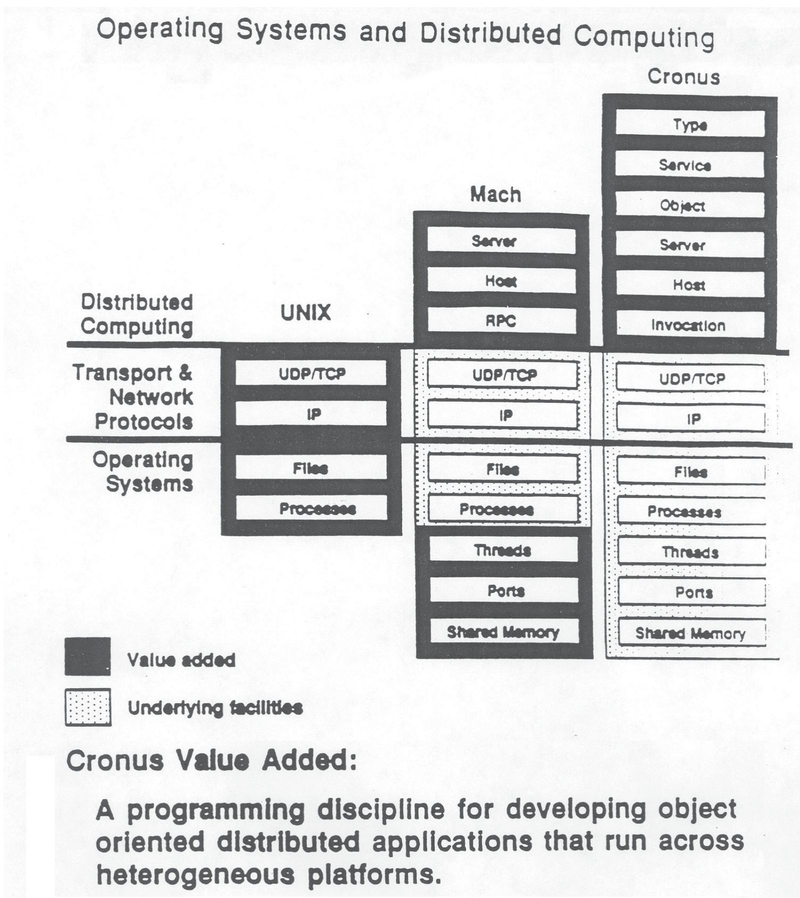


Figure 18.4 System layering concepts, circa 1984.

DCE would be the term used for the standardization effort that the Open Software Foundation would mount beginning about 1986 and go under the name of OSF-DCE. That activity issued a call for technology submissions, to which we offered the Cronus work. However that standard, which was never really successful, stressed a more traditional remote procedure call technology along with some specific set of system services as provided by the main member organizations. It was the difficulty of integrating parts of the adopted solution from a variety of vendors that delayed and likely doomed the OSF-DCE effort. It wasn't until a number of years later that a second attempt at a distributed computing standard (OMG CORBA), this time focusing on distributed objects and this time only specifying interfaces (no implementation). It was that standard which took the ideas first established in practice with Cronus, and raised their profile internationally.

By 1984, the project had succeeded in its initial goals of developing an advanced development model for distributed object computing. It would now grow to place emphasis on a number of related, concurrent technology areas, and integrate them with the evolving distributed object computing base. Strategically, this was part of our belief that the distributed computing environment or middleware was at the intersection of and an integration vehicle for a number of (to date) independent computer science technology disciplines. It was through the middleware perspective of constructing multi-platform solutions that the programming, language, communication, operating system resource management, data base, fault tolerance, and security agendas all seemed to intersect and needed to be combined in a coherent way to meet the needs of application developers in a more unified way across the various platforms used. Accordingly, Schantz developed a plan for the Air Force sponsors to incorporate a number of these technology areas into the ongoing investigation, and to tackle the issues associated with building militarily relevant examples using this new middleware technology. This plan was put into place through a series of RFPs for separate projects, in which we typically teamed with specialists in the specific area to integrate the technology with the evolving Cronus base. This sort of teaming was to be a constantly repeated theme, owing in large measure to the virtue of the middleware solutions we were pursuing as an integrating medium.

By this time Steve Vinter had joined the Cronus project to help with this expansion, after getting his doctoral degree from the University of Massachusetts, where he concentrated on the topic of distributed systems resource management. By now, it was easier to find people coming out of universities who already had specific skills and experience in this emerging area.

Integration with Data Base Access and Secure Operation. Cronus served as a starting point for several other R&D efforts which were pursued simultaneously in the mid ’80s. Steve Vinter and Ward Walker led a distributed database effort, in collaboration with Computer Corporation of America (CCA, an early database company) that led to the development of one of the first generic interfaces capable of encapsulating access to a variety of different relational database management systems.[67] A query engine was also developed which allowed Cronus managers to support SQL queries on their objects. Steve Vinter recalls:

> The database integration project was an attempt to merge mature relational database products with object-based distributed systems, supported by server development tools. All were expected to be important components of rich distributed applications, but presenting a programming model and tools support to the application developer that reconciled their drastically different approaches was a challenge.

In 1985, BBN began a collaboration with Odyssey Research Associates (ORA) to add multilevel security to Cronus.[68] The resulting variant of Cronus was the first instance of a secure distributed OS.[69] At first it was called simply “SDOS” but the name was later changed to THETA (“Trusted HETerogeneous Architecture”). Franklin Webber, now a full time BBN consultant, but at the time ORA’s lead project engineer, recalls:

> BBN’s leadership on the project began with Rick Schantz but the baton was passed early to Steve Vinter, who led the work through the early design phase, and soon to Tom Casey who had extensive experience with multi-level secure systems from his Multics background. On the ORA side, I led the security analysis and later

the implementation of SDOS. When we began SDOS, the understanding within the security community was (and still is) that:

1. the amount of code needed to enforce security should be kept to a minimum, and
2. "retrofitting" security to an existing system is hard. Cronus already existed, and its code base was huge. How were we going to add security to that?

The debate ran between two extremes. On one side, some claimed that the best way to minimize the amount of trusted code was not to secure the Cronus code at all but to secure only the heterogeneous OSs and networks on which it runs. This infrastructure below the middleware would then prevent security violations by the middleware and by Cronus applications. The drawback of that approach is that a separate copy of each Cronus object manager would need to be run for each security classification the system used. I championed the opposite extreme, that we needed to be able to build not just a multilevel secure system, but multilevel secure Cronus managers as well. While this approach was more work for us, I believed it necessary for efficiency and practicality. SDOS completely redesigned the Cronus kernel for security, but the SDOS kernel worked with off-the-shelf Cronus managers. We leveraged the existing Cronus tools for generating managers so that developers could plug in code for processing at one security level; the tools would then generate the secure multilevel manager from that input. This approach was an innovation that reduced the risk of writing corrupt object managers.

Productization and Applications. Beginning in 1985 we undertook the effort to transform the now operational Cronus research prototype into a commonly available utility including formalizing the release cycle, recording and fixing bugs and producing new functionality on a fixed and predictable schedule. First Greg Kenley, then Steve Jeffreys and then Jim Berets led this productization effort and system support effort. From time to time BBN toyed with the commercialization of Cronus, but never in a serious way. We did seek to get commercial customers by separating the GOTS (government off the shelf) product from services which we would offer to commercial clients. But this low level effort never bore fruit and just faded away. In the 1986-88 timeframe we led a number of activities to develop or work with DoD groups to help them develop distributed applications for the robust capability we were now fielding. Ken Schroder and Ron Mucci led the development of a C2 Internet Experiment, where elements of C2 programs were supported partly at Rome in New York and partly at BBN in Massachusetts, using the underlying Cronus mechanisms to structure the interactions. This success led to our working with developers at MITRE, the Air Force Electronic Systems Command at Hanscom AFB, and Rome Lab to connect different stages of models for the Strategic Defense Initiative (SDI, or "Star Wars") program to give them an end-to-end simulation across the participating installations. This activity served as the model for an integrated simulation testbed developed under that program. Distributed application engineers at BBN worked with Department of Defense (DoD) contractors and operational personnel to build and deploy systems such as CASES, TARGET (Theater-Level Analysis Replanning and Graphic Execution Toolbox), and DART,[70] and were able to tackle significant operational issues for distributed command-and-control applications. The Capabilities Assessment and Evaluation Systems (CASES) developed for DARPA and the U.S. Navy was initially developed on a Symbolics Lisp Machine and used Cronus to access legacy FORTRAN models running on a variety of high performance computing platforms. CASES was later ported to Unix, where Cronus was also used to provide the interface between a C/Motif GUI and the Common Lisp application, and to store plans and supporting model parameters. The Dynamic Analytical Replanning Tool (DART) developed for the U.S. Transportation Command was one of several applications that used Cronus primarily to interface between a Common Lisp application and an Oracle relational database. It was used to support Operation Desert Shield in 1990. Mike Dean recalls some of those early DoD applications of Cronus:

> Integration across heterogeneous platforms motivated much of the application use of Cronus. Cronus fully embraced heterogeneity, allowing each component of a distributed application to reside on the platform best suited for its execution, in a day when heterogeneity meant more than supporting both Windows Me and Windows XP. The U.S. Navy, through its Naval Ocean Systems Center (NOSC),[71] also had a distributed computing research program, and soon became an active user and supporter of Cronus. They were particularly interested in tactical data link interfaces, data base integration, replication, and wide area distribution capabilities for Navy tactical applications, and later funded the development of the C++ implementations of Cronus and Corbus.

With the Air Force and Navy Labs both committed to using Cronus, it wasn't long before the U.S. Army's Communications and Electronics Command (CECOM, at Fort Monmouth N.J.) did so as well. This resulted in a Joint Directors of Laboratories (JDL) Tri-Service Distributed Technology Experiment using Cronus as a testbed for wide-area information sharing over broadband IP/TCP networks (initially a 2 Mb/sec satellite link) connecting the primary Command and Control laboratories for each of the military services. Each service maintained its own data, and replicated it to another site for reliability. These experiments were a small step toward interoperation of multi-service systems. Along with conducting various tests and evaluations, we trained DoD contractors and university engineers, programmers, and system management personnel to use the new technology. Jim Berets was the key BBN person responsible for organizing and leading these courses. Subsequently, the systems were incorporated into advanced concept demonstrations by other DoD and DARPA contractors. Universities and industrial research labs also synthesized the results with their own technical investigations, further extending middleware's use and refinement. Over time, successes such as these, in distributed object computing, led to several large-scale, distributed integration programs making world wide military command-and-control operations more responsive and cohesive. These activities were carried out largely from the BBN San Diego office, which specialized in advanced technology transitions.

CORBA. By the start of the 1990s, capabilities like those of Cronus began to appear in commercial products from major companies such as Digital, IBM, SUN Microsystems, and Microsoft, as well as from small startup firms. But because no two systems were alike, operating between them was extremely difficult. To remedy the situation, commercial vendors and users of distributed object technology joined forces to set acceptable industry standards. They formed the Object Management Group, and established the Common Object Request Broker (ORB) Architecture, or CORBA. Their grassroots efforts attracted a great deal of interest, first from vendors, and then from users who saw it as a way to voice their requirements for the newly emerging commercial offerings. One of the main contributions of OMG was in standardizing the terminology around distributed object computing, and establishing standards for interfaces. Implementations were left completely to vendors. In time, Cronus was adapted to the CORBA standard and given the new name Corbus (Cronus Orb, suggested by Ward Walker). The right to use parts of the Corbus technology was eventually sold to Visigenic Software Corporation (now part of Borland), a startup ORB vendor, with key features of Corbus to be integrated into the commercial VisiBroker ORB. This would signal the end of this thread of activity for us,[72] with these ideas firmly planted and generally available from commercial vendors (in a slightly inferior form). Yet another example of BBN bringing ideas to real implementations, nurturing them during incubation and dissemination to early adopters, and departing shortly thereafter with others forming an industry base, and reinforcing the relatively large time scales (~20 years) which it takes for new ideas to firmly take hold. This pattern has been good for innovation, good for economic development, and a lot of fun to boot.

Beginning in 1995 we began a new area of R&D investigation, through a new architectural layer, building upon the now established COTS base. This would concentrate on middleware for managing end-to-end Quality of Service for these distributed object computations, using runtime adaptation techniques to accommodate the changing environments which the Internet had become, with mixtures of high speed and low speed, wired and wireless interconnection, and vastly different sized and capability platforms participating in the new applications. The new project that emerged from this investigation, QuO (for Quality of Service for Objects[73]), is ongoing and thriving, so it's still too early to place it into a historical perspective.[74]

18.4 Internet application focus: 1981-1995

In 1981, Harry Forsdick and Bob Thomas successfully began a major activity for DARPA to investigate how to utilize the emerging new computational capability, capitalizing on having the interconnected Jericho workstations as a basis and laboratory. In retrospect, this initiated a 15-year march forward on projects that built on the significantly improving network and computational base in advanced Internet applications, focusing on what people use and see, rather than the underlying Internet infrastructure and communications.

By 1995, and with the Internet becoming a phenomenon, Forsdick developed another concept: a Personal Internet Newspaper (PINpaper), which was a Web- and agent-based information discovery, filtering, organizing, and presentation system. It allowed customized "newspapers" to be assembled and delivered to your electronic mailbox daily. In 1996, BBN licensed PINpaper to CMGI, which then formed an Internet startup, Information Publishing, based on this technology. Harry left BBN to pursue this start-up opportunity, having sustained over 15 years a steady stream of innovative Internet-based multimedia distributed applications.[75] In the following four subsections, Harry Forsdick recalls these years through the series of discrete projects he was involved with from 1981-1995. He says: "The projects I worked on are easy to describe since many people today use descendents to these early applications that used the Internet."

Diamond, multi-media mail system

The Research in Distributed Personal Computers project, was known for developing a multimedia electronic mail system named Diamond.[76] Our sponsor at DARPA challenged us to come up with an example of the type of application you could build with a collection of distributed personal computers connected by a local area network. Diamond was designed to share some of the ideas and infrastructure being developed simultaneously by the Cronus project, another BBN project whose goal was to create the infrastructure needed to support applications running on multiple computers distributed around a wide area network.

As luck would have it, we also were encouraged to think about how we could improve upon text e-mail. Although today we take for granted e-mail containing multiple font styles, colors, tables, images, etc., in 1982, e-mail was limited to ASCII text. Our challenge was to push the envelope to see if we could do better. Rather than think incrementally and add a new media type, say images, to text e-mail, we went for broke. We focused on the document as being the item that needed to be generalized. Our documents needed to combine multimedia objects by embedding them in structuring objects that organized that content. Figure 18.5 is an example multi-media message from an early design prototype. This was a huge departure from existing text e-mail

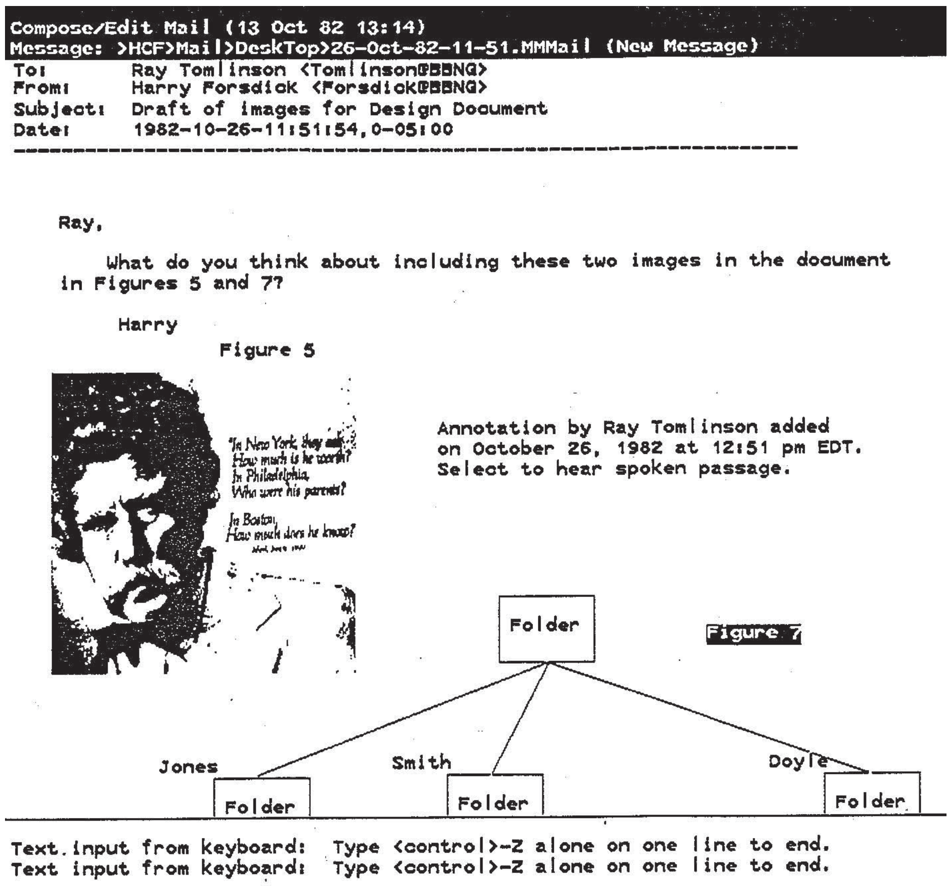


Figure 18.5 An example of an early-concept Diamond multi-media document

where the structure and content were simple and intertwined. We were fortunate to be influenced in this direction because we were working with the Cronus protocols that had the notion of structured data streams. As it turns out, this embedding of objects within other objects has been the subject of several patent disputes in recent years for which Diamond and Cronus have been cited as prior art in successful defenses against accusations of patent infringement. By the end of the project in 1985, the Diamond document editor supported full integration of text, line drawing graphics, images, spreadsheets, and voice in one document. You could write a multimedia document without interrupting your train of thought by switching to another editor to create a figure or spreadsheet. This predated the limited all-in-one systems you see today. In a similar prediction of the future, Diamond documents were stored in a document store — a server like today's IMAP e-mail servers. The editor and document manager communicated with the server using the Cronus protocols.

At the end of the DARPA sponsored research project,[77] BBN decided to invest further in the development of Diamond and called it BBN/Slate. All of the distributed system support was removed to make Slate a stand-alone product[78] and most of the work that was done either improved on features that were already in Diamond or improved the reliability of the system. One notable unique improvement was the addition of multilingual text — i.e., the ability to represent multiple writing scripts in one document. As usual, we went for broke and not only integrated alternate character based writing systems (e.g., Cyrillic) but also symbol based (Hangul) and right-to-left script based writing systems (Arabic). Although BBN/Slate was not a commercial success for BBN, it showed what could be done, which after all, was why we worked at BBN.

Shared workspace conferencing

By 1984, the original Diamond DARPA R&D project was winding down.[79] During the Christmas break time in 1984 I decided to take the Diamond multimedia components (text, graphics, images, ...) and see if I could recast them into a different application — one that took advantage of the real-time communication capabilities of the local area network. The idea was to allow people at two workstations to collaborate over the editing of a single document. This turned out to be a rousing success and caught the attention of a lot of people at BBN as well as at our sponsors. I wrote about this in "Explorations in Real-time Multimedia Conferencing," a paper which it turns out has been cited as prior art in several patent infringement defenses.[80]

When I came back from presenting this paper, Terry Crowley, the lead developer at BBN in multimedia applications, ripped apart the rapid prototype programming efforts and over the next year or so made an infrastructure that came to be known as MMConf. The system that Terry developed intercepted input events (keyboard key transitions, mouse movements) and output events (operations that changed the display) and distributed them to other conference participants. A floor control mechanism based on tokens was used to resolve simultaneous activity and various policies about how to use this mechanism ("raise hand, be recognized," "interrupt at will, take the floor," etc.). We described this system in MMConf: An Infrastructure for Building Shared Multimedia Applications.[81] Subsequent to this work, a lot of people regained interest in the topic of real-time collaboration including many of our competitors in the government contractor community. Of course, Douglas Engelbart had demonstrated many of these ideas in 1975 with the now famous NLS demonstration at the Fall Joint Computer Conference in 1968 and an associated paper written in 1975, "NLS Teleconferencing Features."

We used the MMConf conferencing layer underneath a number of different applications of interest to BBN and to our government sponsors. In addition to Diamond and BBN/Slate, there is one notable application, Shared Map Planning, that illustrates the advantages of customer centered application development. In the late 1980s, I was visiting an Army facility at Fort Leavenworth, Kansas. After showing an officer how Diamond could be used in conferencing mode, I sat down with him and asked what other uses of this underlying technology could he imagine. Immediately he went over to the wall, hung up a map, took a piece of acetate and some markers, and quickly drew up a battle plan using a variety of military standard symbols. At this point he paused and said to me that the standard way of communicating this kind of plan was to roll up the acetate sheet and have a courier take it to another person. This 10 minute discussion was the genesis of both the Shared Map Planning application and the Stand Up Display (see below).

Paul Milazzo implemented all of the features the Army officer described to me into the Shared Map Planning application (map, drawing free hand sketches and standard military symbols on an overlay). Since this was implemented from the start on top of the MMConf infrastructure, conferencing "came for free." This allowed planners at multiple locations to collaborate over the development of a battle plan. The ideas apparent in Shared Map Planning led to the creation of an entire DARPA program known as Distributed Collaborative Planning.

Desktop video conferencing

We didn't know that you couldn't send video from one workstation to another, in software just using the workstation's processor, including voice with echo suppression. Sure you might be able to do this over a local area network, but it would never work over a wide area network — or so the experts told us. When one famous participant in the Internet community saw Picture Window working, he said that this would cause the Internet to "melt down." Paul and I got the idea for PicWin in response to our experiences working on the DWSNet (Distributed Wargaming System Network) project which had a hardware-based video conferencing system. We felt that the requirement for reserving time (and bandwidth) to use the system stifled spontaneous collaborations. Rather, we wanted to be able to pop up a video conferencing window on our workstations in our offices whenever we wanted. This was the motivation behind PicWin, the first Internet-based desktop video conferencing system.

Paul Milazzo was the brains behind all of the programming for the version of PicWin that ran on Sun Workstations. He implemented a simple changed block video encoding algorithm that we could execute quickly on the processors of that era. One of the other interesting aspects of PicWin was Carl Howe's focus on making PicWin a product from the start: this work was all funded with BBN money and so we didn't have to do our usual thing: first convince an outside sponsor, wait for funding and then build it. This is not to say that PicWin was a huge success. As usual, the idea was a lot more compelling than our "word of mouth" marketing and sales efforts. Ideas, prototypes and early products was what BBN was all about. However, there were many people who copied PicWin, most notably, CuSeeMe. People at Cornell bought a single copy of PicWin and applying the rule "if you see that something can be built, building the second version is much easier," they came out with a free version of desktop video conferencing that ran on Macintoshes, a much more affordable platform than our Sun workstation platform. CuSeeMe was commercialized, then sold around during the .COM meltdown and is still being sold and used. For a timeline of video conferencing technology, with several mentions of the work we did on desktop video conferencing, see "A history of video conferencing (VC) technology."

Internet mail/news/web server

Today, we take the Internet for granted, or at least compared to 10-15 years ago. Then, it took a wizard to make Internet things work. I came up with the idea that we should be able to put the three most popular Internet services, e-mail, web and news into one hardware box and sell it as an appliance. As it turned out, the Education Department at BBN was getting started on Copernicus, a project aimed at putting such services into schools. This was the perfect match. We developed the server capabilities in a pre-configured box and the Education Group developed the GUI front end that ran on a Macintosh and administered the server. The idea was a great one, and has been imitated many times since in a variety of flavors, but as they say, the "devil is in the details." The only ubiquitous form of communication in schools (as elsewhere) was dial up. Client Macs and PCs connected to the server using Ethernet. The server would act as a router and send/receive Internet information over this slow and somewhat unreliable path. The reliability of a server was dependent on being able to make and break these dial-up connections: an imperfect solution, at best — but this was the job of the appliance: to take the onus out of connecting to and getting services from the Internet.

The BBN Internet server is a prime example of an idea that was developed before the underlying technology and infrastructure was there to support it. That didn't stop us from trying make up for the missing pieces by applying our engineering expertise to the problem. Today, you can build such a device in a package the size of a Linksys Router, running Linux inside with a boatload of memory. Such "residential gateways/servers" are starting to appear in a variety of forms. One particularly interesting one is Vibe, a software package that runs on a PC and provides a complete audio, video, image, and file serving appliance aimed at the consumer market.

Video information server

Paul Milazzo was fascinated by and the master of all things Video. In the early '90s a lot was starting to change in the transition from analog video to digital video. During this awkward time a variety of forward looking devices were coming out in the consumer market. One that caught our attention was an affordable video disk recorder. With this device as the inspiration we started to think about a Video Information Server: a server that could be used to capture, store, index and playback video clips. We knew that analog's days were numbered, and that basing our server on analog technology was a dead end, but working out the details of the functions of such a server prior to the appearance of affordable digital video storage and transmission capabilities seemed like a good idea. And it was: this device, although mired in the analog world, showed what would be in digital video servers 10 years later, both on the Internet as well as in such consumer devices as TiVo and Replay TV. The server had a scheduling agent which allowed unattended recording of programs. We ran analog video distribution wires back to our offices and built clients that would allow us to record video on our workstations. One of the more clever things we did was to hook a closed caption decoder to the incoming analog video signal and pour the resulting ASCII text captions into a database that was keyed to the position in the video. For those programs that had closed captioning, this allowed us to search a text database for video clips containing search patterns in their dialog.

Once we had a handle on the video clips in a form we could search, the integration of video media into other systems we had developed was easy. So, for example, we could send references to video clips on the server as part of BBN/Slate messages. In addition, we hooked up the PINpaper (see below) so that it was possible to build a topic that searched the Video Information Server's closed caption database for clips containing keyword expressions and filter the clips so that just the clips about a particular topic would be selected.

Today with digital video flooding the Internet and the living room, our work looks pretty primitive: however, as with many other projects at BBN, many of the possibilities of what you might do with stored video were explored in this early system.

Wireless wearable computers[82]

With the computational and networking parts getting smaller, faster and cheaper, around 1993 we proposed to DARPA a project to develop a wearable, networked computational capability. It would include personal sensors, personal communication devices, and anything else electronic someone might likely carry around. The military would be very interested in this sort of technology for the dismounted soldier and for small teams. The key technologies we developed under this project we called Pathfinder, were a personal local area network that could be woven into a wearable vest, and software permitting the attached devices to communicate easily among each other, and with external entities (e.g., other wearers, a base station). We developed a working concept, including trials with the Marines. Battery size and battery life was one of the impediments to effective use, in addition to (usual by now) being too early in the life cycle for sustained viability. In 2003, such devices are being sold, albeit for the electronic eccentric, with designer fashions for the vest!

18.5 Conclusion

In many ways, the years from ~1970–1990 were golden years for BBN and computing technology in general and distributed computing in particular. The brand new rapidly evolving and expanding networking technology opened up a new sense of what was possible and feasible. In almost every direction we turned, for many years, we were stepping into something brand new, never before (sometimes) tried and certainly not made usable and put into practice. It was an exciting time to be a Computer Scientist, especially one focused on the new ideas associated with getting collections of people and their machines working together in a much more intimate fashion.

In retrospect, as measured by what the computing landscape looks like now, in 2003, we were wildly successful in a manner which was beyond our understanding while it was ongoing. It all seemed like so much fun at the time, and certainly had elements of "playing" as much as if not more of "working." Look what we could do, and after doing it, look at what else we could do, on and on until it got to be serious business and/or lots of people starting doing it as well. From e-mail, to networked services, to advanced middleware infrastructure, to collaborative Internet applications, to large scale virtual reality simulations, we were there at the outset and BBN people played major roles in establishing the roots, shaping technical directions and popularization of the ideas. As measured by creating sustained business from these technical innovations, we were less successful, uniformly across the board.

All of that probably says a lot about the BBN culture and the people who shaped it and were part of it, as well as the (government] organizations that repeatedly had the confidence in us to sustain us. Many times we felt as if our lot in life was to develop industries for others to populate. But that's not too shabby either. What we take for granted today in terms of connected computers and interoperability and higher level software for networking, and applications specifically about integrated, connected electronically facilitated or mediated interactions across wide distances, wasn't always so (although to those coming of age now, it certainly appears so). In this paper, and in others of the collection of papers, we have tried to retrace many of the steps we took over the years to help propel going from a few connected computers to a rich (and still growing) set of distributed computing capabilities (some very good, some still very primitive, and some potentially dangerous), and do so before the memories fade into the recesses of the minds of those who participated.

This article is dedicated to all of those BBNers who helped make it happen. Its been a truly amazing ride, with many interesting stops along the way.

Acknowledgments

A number of people helped considerably in putting together this article. Dave Walden has been of great help and assistance to all of the authors in this series. Harry Forsdick, Bob Thomas, Mike Dean, Steve Toner and Jim Calvin provided a good deal of material about what they remembered, and helped to shake loose some of the cobwebs in the minds of what others, including the author, remembered. I directly quoted Harry

Forsdick and Bob Thomas, in particular, at considerable length. Ray Nickerson provided insight and encouragement, as well as reviewing drafts. Mike Dean, Joe Loyall, and Ginny Travers reviewed drafts and also made numerous suggestions for improvements. Many people including Ray Tomlinson, Dan Massey, Jerry Burchfiel, Steve Vinter, Franklin Webber, Al Nemeth and others I may have inadvertently forgotten to mention, provided additional remembrances.

Notes and References

1. A “host” computer was a computer attached to the ARPANET (such as an IBM 360, MIT’s Multics, or BBN’s TENEX) that used the network for communication among computational processes running on the hosts.

2. See also D. Miller and J. Thorpe, “SIMNET: The Advent of Simulator Networking,” *Proceedings of the IEEE*, August 1995.

3. L. Roberts and B. Wessler, “Computer Network Development to Achieve Resource Sharing,” *AFIPS Conf. Proc.*, Vol. 36, 1970 SJCC.

4. D. Bobrow, J. Burchfiel, D. Murphy and R. Tomlinson, “TENEX, A Paged Time Sharing System for the PDP-10, *CACM*, Vol. 15, No. 3, March 1972.

5. RFCs 158, 172, 438, 495, etc.

6. Katie Hafner and Matthew Lyon, *Where Wizards Stay Up Late*, Touchstone imprint of Simon & Schuster Inc., New York, paperback edition, 1998. From more about the development of email, see “The Evolution of ARPANET e-mail,” Ian R. Hardy, History Thesis Paper, University of California at Berkeley Spring, 1996, available at `www.ifla.org/documents/internet/hari1.txt`, Chapter 19 of this book, and Craig Partridge, “The Technical Development of Internet Email,” *IEEE Annals of the History of Computing*, vol. 30, no. 2, April-June 2008, pp. 3–29.

7. This extensive quote taken from an E-mail message from Bob Thomas to Rick Schantz, with subject “Help with history,” dated 03 Sep 2003.

8. R. Thomas, D. Henderson, “McROSS, Multi-Computer Programming System,” *Proc. AFIPS 1972 Spring Joint Computer Conf.*, AFIPS Press, Vol. 40, pp. 281–294,1972.

9. It’s interesting to note that Telneting back to the same host you were connected to became a favorite technique for multi-tasking even through a single host. This was a special case of using Telnet, which was an especially useful tool for debugging cooperating processes.

10. This protocol to handle automatic reconnection, although somewhat cumbersome due to the nature of the connection setup, was used extensively in developing some of the early experimental TENEX services, and was later improved upon by Rick Schantz as reported in RFC 671, “A Note on Reconnection Protocol,” December 1974.

11. Packet (or message) switching was in contrast to circuit switching, the technology underlying the telephone network and the dominant technology of the day.

12. This theme will be repeated later as well, and in fact, the eventual replacement for NCP, became a suite of protocols, IP/TCP, one message oriented, the other connection oriented.

13. C. Carr, S. Crocker, and V. Cerf, “Host/Host Communication Protocol in the ARPA Network,” *Proceedings of the AFIPS 1970 Computer Conference*, Vol. 36, AFIPS Press, Montvale, NJ, pp. 589-597.

14. “A System for Interprocess Communication in a Resource Sharing Computer Network,” July 7, 1970.

15. August 3, 1970, same title.

16. “A System for Interprocess Communication in a Resource-Sharing Computer Network,” *Communications of the ACM*, Vol. 15, No. 4, April 1972, pp. 221-230.

17. Same title, *Proceedings of the Fourth Hawaii Conference on Information and System Sciences*, January 1971, pp. 640-642.

18. *Advances in Computer Communications*, W. W. Chu (ed.), Artech House Inc., 1974, second edition 1976.

19. R. D. Bressler, "Interprocess Communication on the ARPA Computer Network," MIT Civil Engineering, MS Thesis, May 1971.

20. One of the developers of BBN's Lisp systems and of TENEX and mentioned extensively in chapters 4 and 21.

21. Richard E. Schantz, "Operating System Design for a Network Computer," Computer Science Department, State University of New York at Stony Brook, PhD thesis, May 1974; see also "An Operating System for a Network Environment," E. Akkoyunlu, A. Bernstein and R. Schantz, *Proceedings of the Symposium on Computer-Communications Networks and Teletraffic*, Polytechnic Press, New York, April 1972.

22. D.J. Farber and K.C. Larson, "The Structure of a Distributed Computing System Software," *Proceedings of the Symposium on Computer-Communications Networks and Teletraffic*, Polytechnic Press, New York, April 1972.

23. Comparisons of these early investigations into network centric message passing can be found in Eralp Akkoyunlu, Arthur Bernstein, and Richard Schantz, "Interprocess Communication Facilities for Network Operating Systems," *IEEE Computer*, June 1974.

24. Peter H. Salus, *Casting the Net: From ARPANET to Internet and beyond*, Addison-Wesley Publishing Company, Reading, MA, 1995.

25. R. M. Metcalfe, *Packet Communication*, With a Forward by Vinton G. Cerf, Peer-to-Peer Communications, Inc., 1996.

26. R. H. Thomas, "A Resource Sharing Executive for the Arpanet," *AFIPS Conf. Proc.*, Vol. 42, 1973 SJCC.

27. R.H. Thomas, "JSYS Traps — A Tenex Mechanism for Encapsulation of User Processes," AFIPS Conf. Proc., Vol. 44, 1975 NCC.

28. See, for example, P. Johnson, R. Schantz and R. Thomas, "Interprocess Communication to Support Distributed Computing," *ACM SIGCOMM-SIGOPS Interface Meeting on Interprocess Communications*, March 1975.

29. S. Ornstein, F. Heart, W. Crowther, S. Russell, H. Rising, and A. Michel, "The Terminal IMP for the ARPA Computer Network," *AFIPS Spring Joint Computer Conference Proceedings*, Vol. 40, June 1972.

30. This issue of access control and accounting had been around for a while. In April 1971, Bob Kahn, then part of the BBN ARPANET team (and later heading up the office at DARPA responsible for the ARPANET and indirectly still pursuing these issues) wrote RFC 136, titled "Host Accounting and Administrative Procedures," which shows early concern for issues in controlling and accounting for resource usage in the emerging communication utility.

31. See also R. Schantz, "Protocols for Utilizing Redundant Processes in a Computer Network," *Proceedings of the 5th Texas Conference On Computing Systems*, IEEE Computer Society Publications Office, Austin Texas, October 1976, pp. 55–65.

32. "An Operational System for Computer Resource Sharing," B. Cosell, P. Johnson, J. Malman, R. Schantz, J. Sussman, R. Thomas, and D. Walden, *Proceedings 5th ACM Symposium on Operating Systems Principles*, ACM Special Interest Group on Operatings Systems (SIGOPS), November 1975, pp. 75–81.

33. Bob Thomas recalls an interesting bug in our early systems: "The TIP news (`@n`) was supported by multiple TENEX servers that started up an RSEXEC process to service a user. The TIP responded to the `@n` command by broadcasting requests for service to the set of known servers. Early after getting this to work we had a situation where we had a fast but buggy TENEX server which almost always was the first to respond, but which almost immediately terminated the session before providing any service. We fixed the buggy TENEX and I think Bernie [Cosell] modified the TIP to recover, but we learned fastest isn't always best."

34. The terminal interface processor (tip) concept was later renamed terminal access computer (tac) and tacacs was the tac access control system or tacacs.

35. S. Crocker, "The National Software Works: A New Method for Providing Software Development Tools Using the Arpanet," in *Proceedings of Meeting on 20 Years of Computer Science'*, Consiglio Nazionale delle Richerche Instituto Di Elaborazione Della Informazione, Pisa Italy, June 1975.

36. Jon Postel and James White, RFC 674, "Procedure Call Protocol Documents, Version 2," December 1974.

37. R. Schantz, RFC 684, "Commentary on procedure calling as a network protocol," April 1975.

38. Note the emergence of the message passing vs. procedure call debate, as a follow on to the previous message oriented vs. connection oriented controversy. This was also another early sign of merging a network communications point of view with a software engineering point of view.

39. After entering the project, we largely designed the operating system-like infrastructure functions for interprocess communication and running programs on TBHs, while MCA largely designed the resource (Works) manager and file transfer functions. SRI largely designed the initial user interface. BBN built TBH components for TENEX, MIT for Multics, and UCLA for OS/360. Over time, the role of SRI became diminished, although some of the ideas they were contributing remained key, especially formalizing the interactions between components.

40. For a more comprehensive discussion of NSW see Richard E. Schantz and Robert H. Thomas, "A Technical Overview of the National Software Works," RADC Technical Report RADC-TR-83-80, March 1983.

41. R. Thomas, R. Schantz, S. Schaffner, R. Millstein, and P. Johnson, "MSG: The Interprocess Communication Facility for the National Software Works," BBN Report No. 3287, January 1976. Note the common theme of another attempt at reshaping the underlying connection based substrate with a message orientation, as discussed earlier.

42. R. E. Schantz and R. E. Millstein, "The Foreman: Providing the Program Execution Environment for the National Software Works System," BBN Report No. 3442, January 1977.

43. The NSW project team grew to support this development activity, and over its lifetime in addition to Schantz and Thomas included Henrik Lind, Steve Swernofsky, Steve Toner, Dan Massey, Craig Cook, Frank Ulmer and Don Erway.

44. Two of the key proponents at RADC were Pat Baskinger and Richard Robinson.

45. Steve Toner recalls: "Other Fenway House alumni who joined BBN full time included Dave Mankins, Walter 'Doc' Urbaniak, Dan Tappan, Bill Cote, Tom Calderwood. There may be others but those are the ones I remember. Other Fenway House Tenex operators (who did not go on to become full-time BBN employees) included Mike McMahon, Mike Travers and Tom Ricciardi."

46. P. R. Johnson and R. H. Thomas, "The Maintenance of Duplicate Data Bases," RFC 677, January 1975.

47. Created by Martin Richards, BCPL was a precursor to the more ubiquitous C language, whose popularity among researchers was significantly increased by availability of interfaces to ARPANET communications.

48. See, for example, Harry Forsdick, Richard Schantz and Robert Thomas, "Operating Systems for Computer Networks," *IEEE Computer*, January 1978, pp. 48–57.

49. P. R. Johnson and R. H. Thomas, "The Maintenance of Duplicate Data Bases," RFC 677, January 1975.

50. R.H. Thomas, "A Solution to the Concurrency Control Problem for Multiple Copy Data Bases," *Proc. IEEE Spring COMPCON*, San Francisco, February 1978.

51. Robert Thomas, Richard Schantz and Harry Forsdick, "Network Operating Systems," Rome Air Development Center Technical Report RADC-TR-78-117, May 1978 (also available as BBN Report No. 3796).

52. Harry Forsdick, William MacGregor, Richard Schantz, Steven Swernofsky, Robert Thomas and Stephen Toner, "Distributed Operating System Design Study: Final Report," BBN Report No. 4674, Prepared for Rome Air Development Center, May 1981.

53. Bob Thomas recalls the difficulty we had in eventually winning this activity. "A day or so after we learned that we had initially lost the contract to a much less expensive proposal by Computer Sciences Corporation, I happened to be in Rome for a contractors 'show and tell' conference and a group of us went out to dinner with Tom Lawrence, one of the RADC program managers. I remember standing outside a Rome New York restaurant after dinner in subfreezing temperatures with only a very light jacket talking privately for about an hour with Tom about the contract and why we had lost. After some prodding he said that the primary reason was that the BBN bid was just too expensive. I explained to him that those were our ideas they were asking for, and reminded him of the good work we had done for them in the past, and how I thought that RADC had made a serious mistake. I said that I thought we could lower our bid substantially - not knowing at the time, in fact, whether we could or not. Some time later we heard that the award to CSC was not finalized and the RFP was recompeted. With everybody wiser by now, we eventually won the contract on the third try. I certainly hope my standing in the freezing weather of upstate New York helped the RADC program manager better see the value in having us continue working on these ideas."

54. Taken from "Cronus, A Distributed Operating System," R. Schantz et al, Interim Technical Report #1, for the period 8 June 1981 through 30 June 1982, RADC Technical Report TR-83-236, November 1983.

55. R. Schantz, R. Thomas, G. Bono, "The Architecture of the Cronus Distributed Operating System," *6th International IEEE Conference on Distributed Computing Systems*, Cambridge, MA, May 1986; R. Gurwitz, M. Dean, R. Schantz, "Programming Support in the Cronus Distributed Operating System,"*6th International IEEE Conference on Distributed Computing Systems*, Cambridge, MA, May 1986.

56. These include Rick Schantz, Bob Thomas, Bill MacGregor, Steve Toner, Girome Bono, Mike Dean, Steve Vinter, Ken Schroder, Jim Berets, Don Allen , Chris Barber , Hunter Barr , Marcus Barrow, Paul Bicknell , Chuck Blake, John Bowe , Ed Burke, Tom Casey , Natasha Cherniack Westland , Jon Cole, Andres Echenique, Chantal Eide, Rick Floyd, Harry Forsdick, Andy Gerber, Steve Geyer, Rob Gurwitz, Mort Hoffman, Carl Howe, Kathy Huber, Steve Jeffreys, Penny Karr, Greg Kenley, Karen Lam, Ken Lebowitz, Sam Lipson, Dick Mackey, Dave Mankins, Ron Mucci, Paul Neves, Mark Nodine, Craig Partridge, Sue Pawlowski Sours, Ken Pogran, Kobita Rajan, Rich Salz, Rich Sands, Dan Tappan, Huong Ton, Vic Voydock, Ward Walker, Bob Walsh, Bob Willis, Mark Woodworth, and Ben Woznick.

57. In addition to preparing documents for that archive, the Cronus project instituted from the beginning a series of DOS Notes, patterned after the successful RFC series started with the ARPANET. These notes (over 100 of them) were the recorded history of the project, including design studies, discussion of technical issues of the day, recorded decisions, and presentations of the material at various reviews. They serve as a base of information for occasionally reminding us what the computing environment and issues of concern were 20 years ago.

58. BBN Report 5041, "Cronus, A Distributed Operating System, Functional Definition and System Concept," June 1982.

59. This list of innovations was adapted from "Cronus, A Short Summary," an unpublished note by Mike Dean, 1998.

60. "Cronus, A Distributed Operating System, Functional Definition and System Concept," BBN

Report 5041, June 1982. Also available as “Cronus Interim Technical Report #1” for Rome Air Development Center, RADC-TR-83-236, November 1983.

61. “Cronus, A Distributed Operating System, System/Subsystem Design,” BBN Report 5086, July 1982. Also available as “Cronus Interim Technical Report #2” for Rome Air Development Center, RADC-TR-255, December 1983.

62. Steve Toner recalls what it was like to be an early adopter: “When we started work on the GCE, there was only one Multibus Ethernet adapter available. This was from Intel, and was fiendishly expensive (about $3,000 as I recall). It also took two card slots. This adapter had a microprocessor (8088?) on one of the borads and communicated with the main CPU through a message-passing interface. Unfortunately, the only documentation for the interface was some PL/C code written for the Intel processor. We had a Motorola 68000, and so I had to try to convert the code to C and deal with byte-ordering issues as well. I never got it to work. Fortunately, around this time Bob Metcalfe had founded 3Com and their first product was a Multibus Ethernet adapter. These were apparently in great demand, but Ken Pogran knew Metcalfe and pulled some strings so we got a couple of boards off the front of the queue. We got the documentation for the boards before the boards arrived, and I think it was only a couple of days after we got the boards that I was able to send packets with them.

63. RFC 824, W. MacGregor, D. Tappan, “Cronus Virtual Local Network,” August 1982.

64. Steve Toner recalls: “I believe the Cronus VLN was the thing that kicked off the whole ARP discussion. I kind of remember that as soon as RFC 824 was issued, there were a couple of people over at MIT (I specifically remember Noel Chiappa, but it appears that Dave Plummer was also involved) who started discussing it and came up with their own counter-proposal (basically) that was ARP, in RFC 826. Now, I think once that was out we adopted it and never actually used the VLN that we had defined. I remember implementing a rudimentary ARP for the GCE (rudimentary because it never flushed its cache), but I don’t remember implementing RFC 824.”

65. See RFC 947, K. Lebowitz and D. Mankins, “Multi-network Broadcasting Within the Internet,” June 1985.

66. After many years at BBN and after much prodding, Girome eventually earned his Harvard bachelors’s degree.

67. S. Vinter, N. Phadnis and R. Floyd, “Distributed Query Processing in Cronus,” *Proc. of the 9th ICDCS*, June 1989.

68. This project was funded by RADC, in particular by Dick Metzger and Emilie Siarkiewicz.

69. Thomas A. Casey, et al.,“A Secure Distributed Operating System,” *Proceedings of the IEEE Symposium on Security and Privacy*, 1988.

70. See, for example, B. Anderson and J. Flynn, “CASES: A System for Assessing Naval Warfighting Capability,” *Proceedings of the 1990 Symposium on Command and Control Research*, June 1990; also available as Science Applications International Corporation Report SAIC-90/1508.

71. Key supporters at the Naval Ocean Systems Center (NOSC, now SPAWAR Systems Center) in San Diego included Les Anderson and Russ Johnston.

72. Although Mike Dean reports that at least one distributed Cronus application was still running operationally at several military sites as late as the year 2000.

73. Joseph Loyall, Richard Schantz, John Zinky, and David Bakken, “Specifying and Measuring Quality of Service in Distributed Object Systems,” *Proceedings of the First International Symposium on Object-Oriented Real-Time Distributed Computing*, IEEE Computer Society, April 1998, Kyoto, Japan, pp. 43–52.

74. The issues in addressing the problems of dynamic and adaptive end-to-end QoS management which we are undertaking with the QuO activities requires a distributed object computing base as well. Having retired our Cronus activities but never wanting to go backwards, we now have substituted a more modern and currently evolving Corba base for this work in the form of the TAO (The Ace Orb) software, developed by Doug Schmidt et al at Washington University and

Vanderbilt. This software is a modern equivalent of our Cronus work, for the lower levels of middleware support, and more focused on the realtime behavior needed for QoS management for embedded environments.

75. Harry, years later, semi-rejoined BBN through the spun-out Genuity ISP subsidiary, as the climate for Internet startups began to disintegrate. He then worked briefly for Level 3 Communications, as a result of its acquisition of the bankrupt Genuity. He continued to pursue ideas for advanced Internet applications until his retirement.

76. Harry C. Forsdick and Robert H. Thomas, "The Design of Diamond- A Distributed Multimedia Document System," BBN Report No. 5204, November 1982.

77. Probably everyone at one time or another has accidentally sent an e-mail to the wrong person. Bob Thomas recalls a funny incident in this regard:

> When we tried to extend the Diamond Distributed Personal Computer project ARPA decided not to fund it (or to drastically reduce the funding - I don't recall which). In an attempt to understand why and appeal the decision I composed an e-mail for Bob Kahn, who at the time was still head of ARPA IPTO, pointing out all of the good work we had done (Bob T. seemed to do this a lot), asking why we were being cut, and basically begging him to reconsider. I sent the e-mail to `rek@sri`. Unfortunately Bob's e-mail was `rek2@sri`. As it turned out, `rek@sri` was someone on an SRI team that we competed with. How could the head of IPTO and one of the ARPANET pioneers be 'rek2' and not simply 'rek'?

78. This is similar to the evolution of Mach from a distributed platform based on some new directions into a Unix replacement. Perhaps the lesson we continue to learn and relearn is that there is a significant difference between trailblazing and making something that's immediately recognizable and useful.

79. By this time, Bob Thomas had moved on to work with BBN's Butterfly multiprocessor activity (see chapter 21). That subsequently led to activities with the Lightstream high speed switch, based on some of those multiprocessor interconnect ideas. When the Lightstream activity was bought by Cisco, Bob became a Cisco employee, where he remained active in networking projects until his retirement.

80. Forsdick, H. C., "Explorations in Real-time Multimedia Conferencing," *Proceedings 2nd International Symposium on Computer Message Systems*, IFIP, September 1985, pp. 299–315.

81. Terrence Crowley, Paul Milazzo, Ellie Baker, Harry Forsdick and Raymond Tomlinson, "MMConf: an infrastructure for building shared multimedia applications," *Proceedings of the ACM conference on Computer-supported cooperative work*, Los Angeles, 1990.

82. This final subsection is again from Schantz.

Chapter 19

Networked E-mail

Compiled by David Walden

This chapter, compiled from communications with the participants, describes BBN's involvement in the development of networked e-mail

---

The broader story of the development of networked e-mail in the early years is well told in chapter 7 of Katie Hafner and Matthew Lyon's book, *Where Wizards Stay Up Late*.[1,2] This chapter emphasizes BBN's role in the overall story. Craig Partridge, one of the coauthors of Chapter 17 of this book, has also written a more scholarly history of networked e-mail that extends beyond the early days and BBN contributions.[3] (No coverage is given to e-mail activities prior to ARPANET e-mail.)

This section was compiled by Dave Walden with contributions and quotations from many participants in the BBN e-mail story. These contributors are noted throughout the chapter, and their contributions are greatly appreciated. Except for final copy edits in 2010, Walden's compilation effort stopped on August 21, 2003.

19.1 Tomlinson's initial demonstration

Of course, like many innovations (and most Internet innovations), networked e-mail as it exists today has evolved from the efforts of many key contributors over the years. E-mail with a single machine had existed for some time; for example on the CTSS machine at MIT. According to Ray Tomlinson,[4] a program called SNDMSG originated with the Berkeley-developed SDS 940 time-sharing system that BBN was using before TENEX.[5] Tomlinson rewrote SNDMSG for TENEX and started embellishing it in various ways. SNDMSG was used for sending an e-mail within TENEX; the system's Type (a file) command was used to read e-mails.

In 1971, with two TENEX systems available at BBN and with both of them connected to the ARPANET, the possibility of e-mail among users on multiple machines occurred to Tomlinson. Without hesitation he implemented the first instance of networked e-mail between the two TENEX machines.[6] This implementation included SNDMSG as the first network e-mail sending program, the @-sign separator, the business memo format (consisting of lines for To, Subject, From, Date, and CC), the use of the computer's Type (a file) command as a readmail command,[7] and an experimental file transfer protocol (CPYNET) to convey e-mail messages across the network. Today, almost 40 years later, after much iteration and refinement, the outline of Tomlinson's basic implementation model can still be seen in networked e-mail.

Networked e-mail is clearly one of the major components that make the Internet what it is today. Networked e-mail was also the first Internet "killer app" and, when it burst onto the scene in 1971, gave the first tangible indication of how far the Internet might go in becoming the ubiquitous anyone-anywhere-to-anyone-anywhere communication system it has become. E-mail succeeded because it provides interaction at the convenience of the users (they don't have to think in lockstep), but still fast enough for several turnarounds a day (far faster than Post Office mail), supporting a high metabolism of interaction and collaboration. Unlike telephone conversations, it is in a written form and thus creates a record for easy filing and forwarding to other collaborators. Since it is network based, e-mail reaches users worldwide. All of these critical elements were present in the networked e-mail system first demonstrated by Ray Tomlinson.

For his original inspiration and demonstration, Tomlinson received the 2004 IEEE Internet Award (jointly with networked e-mail codifier Dave Crocker), "For their key roles in the conceptualization, first implementation, and standardization of networked e-mail." Tomlinson's early e-mail work (particularly his choice of the at-sign in mail addresses) has also been honored with several other awards.

19.2 A succession of e-mail programs

Many e-mail programs were written by members of the ARPANET community to improve the user interface beyond what Tomlinson provided in his first demonstration. John Vittal, who came to BBN from ISI in 1976, remembers the history as follows.[8] Originally, the TENEX systems ran two programs to send and receive messages: SNDMSG and READMAIL. Next came RD, a set of TECO macros from Larry Roberts at ARPA, that let you selectively read messages from your e-mail inbox. In 1972 Barry Wessler (then at ARPA) started writing a program called NRD (New RD), which was to be a successor to RD, but which was never completed or distributed.

NRD evolved into two e-mail programs, BANANARD and MSG. First, in late 1973 and early 1974, Martin Yonke (then at ISI) and John Vittal got Wessler's code running and called the result WRD for Wessler's RD. Yonke recalls that WRD was only around briefly and was largely Wessler's code with bug fixes but otherwise unchanged.[3] Then Yonke took WRD and changed the interface to make BANANARD, and in parallel Vittal took WRD and BANANARD and made significantly more changes creating MSG.

BANANARD and MSG were the first mail systems on the ARPANET to integrate message reading and creation functions by providing a single user interface; both invoked SNDMSG (as a subprocess) for mail creation. MSG provided a different functionality than BANANARD; specifically, it added a user profile, a more concise user interface, multiple folders for message filing, and the the first instances of the Forward and Answer (reply) commands.[9]

As Vittal remembers, getting the semantics right for the Answer command took some experimentation, which resulted in innovations such as providing options of sending only to the originator of the message or to all recipients, and filling in the subject field with "Re:" the subject of the original message.

The availability of MSG spread by word of mouth and by the mid-1970s it had an active user community of more than 1,000 people. Vittal reports that MSG was never officially funded or supported, other than by him in his spare time. Nonetheless, it clearly had an impact. It went into UNIX and became the starting point for e-mail systems such as MH, MM, and MS. In 1976 Vittal joined BBN, where he continued to maintain MSG mostly on his own time. After the early 1980s, Vittal ceased maintaining MSG, even though it was still in use by a few people as late as the mid-1990s.

Jim Calvin was another BBN person who brought an e-mail program with him when he came to BBN. He says,[10] "I joined BBN in May of 1974 and brought an e-mail program with me I'd done at Case. In 1974 and 1975, I rewrote this program (going from a SAIL implementation to PDP-10 assembler) which became known as Mercury, or HG. I did this on my spare time and actually caused a few headaches for the Mailsys/Hermes guys.[11] HG was much faster until the main mail file had more than ~600 items in it. HG was a full-featured mail program and was used by quite a few people at BBN. It was around into the early 1980s when I was just too busy to keep it going."

19.3 Codification of the e-mail standard

Developing a standard for networked e-mail was a torturous process that took many years and included much (sometimes acrimonious) debate. Different e-mail programs (such as some of those mentioned in the previous subsection) needed different e-mail protocol capabilities. Also, different computer operating systems had more or less difficulty with various aspects of networked e-mail. As the source of TENEX, probably the most popular computer system on the ARPANET in the early days, BBN played a considerable (not always welcome) role in this standardization. (Of course, much work also was done and documented in RFCs and elsewhere by non-BBN people.)

In 1972, the developers of the FTP (file transfer protocol) specification[12] included the possibility of "piggybacking" Tomlinson's networked e-mail messages on FTP, eliminating the need for CPYNET.

In 1973, RFC 561, entitled "Standardizing Network Mail Headers," was published by Abhay Bhushan and Ken Pogram of MIT, Ray Tomlinson of BBN, and Jim White of SRI. Ken Pogran remembers[13] his interest in this standardization effort. He was working on MIT's Multics system and struggling with properly displaying the user who sent a message, date and time sent, and so forth. The Multics e-mail system displayed a message header based on the Multics user IDs, but this was a system process on Multics, not the user who actually sent the message from another site. As a courtesy, the e-mail programs on each ARPANET computer pre-pended to the actual user-generated text some rudimentary header information, but each e-mail program provided this courtesy in a little different fashion. This was OK for some of the more popular computer systems (e.g., TENEX) and e-mail systems (e.g., MSG), which worked relatively consistently with each other. However, in the early days there was only one Multics on the ARPANET, and Pogran did not want Multics to appear "less equal," particularly to the ARPA program managers, who all used TENEX. Thus, Pogran needed some standard that Multics could follow. Abhay Bhushan was better known in the ARPANET community (e.g., as a leader of the specification of FTP) than Pogran, who had recently graduated from MIT; and Bhushan was already in contact with Tomlinson. White was involved because the Network Information Center at SRI wanted to distribute ARPANET documents (e.g., RFCs) via e-mail rather than the postal service and desperately needed a standard.

In 1975, Ted Myer and Austin Henderson of BBN published RFC 680, entitled "Message Transmission Protocol," an improvement on the e-mail protocol.

In May 1977, Ken Pogran of MIT (he joined BBN in 1980); John Vittal and Austin Henderson, both of BBN by that point; and Dave Crocker of RAND published RFC 724, entitled "Proposed Official Standard for the Format of ARPA Network Messages." This assertion of a "standard" was not well received by some in the ARPANET community.[14] Undaunted, in November 1974, Crocker, Vittal, Pogran, and Henderson published a revision of RFC 724 — RFC 733, entitled "STANDARD FOR THE FORMAT OF ARPA NETWORK TEXT MESSAGES." However, RFC 733 didn't end the e-mail protocol debates; in particular, it was incompatible with Vittal's own highly popular MSG e-mail program, according to Hafner's book.

The real "standard" finally was written by Dave Crocker (by then at the University of Delaware) as RFC 822, entitled "STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES" and obsoleting RFC 733. The multiyear effort culminating in this RFC is a primary reason Crocker shared the 2004 IEEE Internet Award with Ray Tomlinson.

Later MIME and other capabilities were added to networked e-mail, but BBN was no longer significantly involved.

19.4 Other BBN e-mail systems

In addition to the BBN systems mentioned in this section, there were also significant e-mail components of CSNET (see Chapter 17), Diamond (Chapter 18), and perhaps other applications systems.

Hermes

According to Jerry Burchfiel,[15] DARPA program manager Steve Walker supported development of a "real" mail system, as opposed to the quick hacks Ray Tomlinson and Larry Roberts had done (SNDMSG, READMAIL, RD, etc.). He funded BBN's development of HERMES, managed by Ted Myer with contributions from Austin Henderson, Ron Brackman, Art Pope, Frank Ulmer, and others. Burchfiel remembers that Walker also funded work at USC ISI. Walker's director at ARPA challenged him to come up with a realistic scenario for transition of this technology into the services, and Walker originated the Military Message Experiment (MME) at CINCPAC, Camp Smith, Hawaii. Both the ISI system and BBN's (early stage) HERMES went out there for two years of testing and evaluation.

Steve Walker remembers[16] that when he got to DARPA, ISI was already working to some extent with people from NRL and CINCPAC on an e-mail demonstration. He became aware of BBN's e-mail work and decided that two approaches might improve the chances of something usable's being produced for CINCPAC. (Al Vezza at MIT also offered an e-mail system for testing without funding from DARPA.[17])

According to John Vittal,[18] sometime in 1975, ARPA funded the Military Message Experiment (MME) to produce an e-mail system that could support multilevel security and priority traffic for the Navy.

In December 1975, John Vittal and BBN's Austin Henderson met at the first e-mail standards meeting in Los Angeles, and Henderson told Vittal that Walker had told BBN to look at MSG so "Hermes could get it right." Eventually, Vittal was offered a job with the Hermes group and joined BBN in July 1976. When Vittal joined the Hermes project, Austin Henderson, Doug Dodds, and Charlotte Mooers[19] were working on the Hermes project, under the leadership of Ted Myer. Jim Miller joined the project next, and Debbie Deutsch joined the project in January 1977. Later Barbara Wagriech joined the project.[20]

In the end, ISI won the MME fly-off (". . . it became obvious," says Vittal, "that ISI's effort[21] was preordained to win the experiment"). Thus, BBN's funding dried up.[22]

Debbie Deutsch remembers,[23] "Hermes attempted to be very flexible/complete compared with its predecessors such as MSG. In particular, it had what amounted to a database capability built into its message store. Plus, it had a template facility to control the display of message fields (presence, order). Hermes had a great many commands with specific names which represented particular combinations of basic commands (such as to display a message) and modifiers (such as a display template). In retrospect, the flexibility/complexity of Hermes' interface made it difficult to approach for new users, and probably worked to its detriment." Users wanted to be able to do e-mail simply. Vittal adds,[24] "[Hermes'] functionality had something for everyone — it really was a research tool to find out what people needed when they did e-mail. However, the defaults were such that it was difficult to use and understand; the system got in the way of *doing* e-mail. Had we had the funding or prescience to run human factors[25] and user interface testing experiments on Hermes, we could have provided a sufficient wealth of design criteria that would have guided e-mail clients to the current day." Steve Walker,[26] however, notes that he personally liked Hermes because it practically could be used as a database management system. He remembers building a system himself in Hermes to handle all the registrations of the First DoD Computer Security Conference.

BBN tried to promote Hermes within the government. According to Deutsch,[27] perhaps the "apex of Hermes deployment came when it was used early in the Carter administration in the Executive Office of the President," says Deutsch. "You would not believe the level of support we gave them. I remember being on call while on vacation. I have hazy memories of Hermes being used when Carter took a vacation trip to the Snake River in Idaho. I have no idea how they connected to the net." Attempts were made to commercialize Hermes. Deutsch remembers when she and Ted Myer visited Telenet, the packet-switching common carrier BBN had founded and partially funded, but they weren't interested. "They felt that e-mail would never be a big thing, since executives wouldn't be caught dead using keyboards or having them in their offices. Since secretaries would be doing all the sending and receiving, what improvement did it offer?" Still, Vittal remembers that eventually Ted Myer left BBN and joined Telenet to try to commercial e-mail.

Intelpost

Julie Sussman was the primary source of information regarding Intelpost,[28] although a little bit of information came from Ray Tomlinson. (In some of the following I paraphrase Sussman and Tomlinson rather than quoting them.) Others participating in the project included Bob Clements and Jim Miller.

In the late 1970s, many people did not yet have access to fax machines, and special delivery was expensive. Thus, the USPS contracted with Comsat to demonstrate a system to scan letters the users brought to a post office and to transmit them to other post offices, perhaps in other countries. Comsat contracted with BBN to do the software for the system.

BBN started work in 1978, coding in BCPL for a PDP-11 and using TCP with routing hard-coded into the software. In June 1979, BBN did a four-node test. The January 1980 brochure for the official "First Day of Transmission" and the October 1980 public Intelpost brochure list, between them, the following countries as participating in the Intelpost system: Argentina, Belgium, Canada, France, Germany, Iran, Netherlands, Switzerland, and the United Kingdom. A June 1980 announcement from the U.S. Postmaster General says the kickoff of service was between Canada and the United Kingdom (not the United States, due to regulatory problems). Sussman remembers that Iran and France wanted to be up first, "but Iran had a revolution and France's PTT (postal and telecommunications) heads were bickering too hard over which half (P or T) was doing this project (since it was both telecommunications and postal service) to actually do anything [August 1979, Datamation]." By September 1981, Buenos Aires was on line for demonstrations.

BBN's software was delivered to each of the sites. Sussman says, "Basically I think we finished, tested, and fixed the software, and configured it for additional sites and for foreign languages (in the operator interface). I think our role was over by the end of 1980, except for delivering a Buenos Aires system in 1981."

InfoMail

In 1980, Dave Walden and John McQuillan wanted to move from networking R&D and consulting to something more commercial (and independent of Frank Heart's division). They talked with BBN president Steve Levy and with Mike Lavigna, BBN's corporate business development person, and wrote a business plan for a commercial e-mail product; as a result BBN started BBN Information Management Corporation with Walden and McQuillan leading it.[29] The e-mail product was named Infomail.[30] Walden served as president of BBN IMC and was "Mr. Inside," managing the day-to-day operation of the business and leading the product development effort; McQuillan served as vice president and was "Mr. Outside," leading the marketing, sales, and customer support.[31]

InfoMail almost certainly was the first multiplatform e-mail system (certainly BBN billed it that way at the time[32]). The hope was that companies and other institutions beginning to adopt e-mail would choose InfoMail because it could run on all of their computer systems. Up until that time, e-mail systems had tended to be machine or operating-system dependent. To support this portability, InfoMail was written in RATFOR, the language preprocessor from the UNIX world that converted a C-like programming language into Fortran; of course, Fortran compilers were available for virtually all computers and operating systems.[33] Version of InfoMail ran under UNIX (for the Digital PDP-11 and the BBN C/70), Digital's VAX\VMS, IBM MVS, and IBM CICS.[34]

Unlike Hermes (discussed above), InfoMail had a succinct set of commands focused on the e-mail task that users seemed comfortable with. InfoMail displayed (primitively, on a terminal screen or page) a desktop, file drawer, and file folders for message management.[35] Nonetheless, while InfoMail system was sold to some companies and institutions, it was not in sufficient volume for this BBN start-up business to be a success. BBN and the ARPANET community were ahead of most of the rest of the world in adopting e-mail.[36] After a couple of years, BBN Information Management Corporation was shut down and its staff and product merged into BBN Communications Corporation. At BBNCC, InfoMail was widely deployed in the Defense Data Network, where it was highly regarded and used for many years.

Notes and References

1. Katie Hafner and Matthew Lyon, *Where Wizards Stay Up Late*, Touchstone imprint of Simon & Schuster Inc., New York, paperback edition, 1998.

2. See also, Ian R. Hardy, "The Evolution of ARPANET email," History Thesis Paper, University of California at Berkeley, spring 1996, available at `http://tinyurl.com/y9tle7a`

3. Craig Partridge, "The Technical Development of Internet Email," *IEEE Annals of the History of Computing*, vol. 30, no. 3, April–June 2008, pp. 3–29.

4. E-mail of September 18, 2003.

5. See page 523 of section 21.2.

6. According to Partridge,[3] Tomlinson deliberately did not go in the direction suggested by Dick Watson of SRI in RFC-196 (July 1971).

7. Later Tomlinson wrote an explicit readmail program in assembly language and still later another version in BCPL. At least the latter was named READMAIL.

8. E-mails of July 28, 2003, and April 8–10, 2010.

9. John Vittal, "MSG: A Simple Message System," in *Computer Message Systems*, ed. Ronald P. Uhlig, New York: North Holland Publishing Co., 1981.

10. E-mail of March 6, 2003.

11. More about Hermes later in this section.

12. Abhay Bushan at MIT, Alex McKenzie at BBN, and others.

13. E-mail of July 30, 2003.

14. Alex McKenzie, in Frank Heart's computer division, with its more primitive computing capabilities than were available in the division doing TENEX, spoke out relatively publicly against the complexities of the proposed standard, which he worried would require a more powerful e-mail program or better computer terminals than those he was using.

15. E-mail of August 7, 2003.

16. E-mail of August 20, 2003.

17. Editor's note: In all likelihood, this was within the Dynamic Modeling Group at MIT, which at that point was J. C. R. Licklider's group, managed by Vezza.

18. E-mail of July 31, 2003.

19. Wife of Calvin Mooers ["Calvin Mooers, the NOL Computer Project, and John Vincent Atanasoff: An Introduction," and "The Computer Project at the Naval Ordnance Laboratory," *IEEE Annals of the History of Computing*, vol. 23, no. 2, pages 50–67], the inventor of TRAC, which had an early (perhaps its first) implementation on BBN's PDP-1 by Peter Deutsch.

20. At the time, BBN was trying to find situations suitable to hire some handicapped people. Barbara Wagriech was deaf and blind. Charlotte Mooers learned to finger talk with her.

21. Named SIGMA and built on top of BBN's TENEX.

22. With the end of Hermes funding, funding for another communications research effort also ended. Starting in about 1977, John Vittal developed a system he called R2D2 (after the circa 1977 "Star Wars" movie — R2D2 stood for Research-to-Development-Tool). The idea was to have programs be transported and executed remotely, with communication back to the originator. John used e-mail as the communication mechanism. This work was documented in a book chapter by Vittal ("Active Message Processing: Messages and Messengers," *Computer Message Systems*, Ronald P. Uhlig, ed., North Holland Publishing Co., New York, 1981) and was the first publication on "active messaging."

23. E-mail of February 11, 2003.

24. E-mail of July 31, 2003.

25. BBN had an outstanding in-house human factors group, as described in Chapter 8.

26. E-mail of August 8, 2003.

27. E-mail of July 30, 2003.

28. E-mail of March 26, 2003.

29. That people from the BBN division where the e-mail work had not previously been done were starting BBN's commercial e-mail activity was a bit distressing to the people working on Hermes in the "e-mail division."

30. Pete Kaiser, Morris Keesan, and Elise Sargent (e-mails of July 30, 2003), John McQuillan (e-mail of August 4, 2003), and Ken Turkewitz (e-mail of August 5, 2003) helped compiler Dave Walden remember the people names and computer systems relevant to InfoMail.

31. The rest of the InfoMail development team that gathered over time included: Curt Sanford (who later was one of the earliest employees of Lucent), Pete Kaiser (who brought IBM experience), Rick Chatranon, Elise Sargent, Ken Turkewitz, Audrey Mack (who brought IBM experience), Morris Keesan, and Bonnie Friedman. Now-renowned U.C. Berkeley professor Randy Katz was part of the development group briefly after he received his PhD, until he concluded a few months later that he was better suited to an academic environment and joined the faculty of the University of Wisconsin. Chris Souter, who had been a saleswoman for IBM (of typewriters, perhaps) was hired to lead the InfoMail sales effort. First Marianne Steiner and later Linda Ridlon managed customer support. Mary Gillis was also involved in business development. Melinda Thedens wrote and producing documentation. Arlene Scherer did customer training. Rob Jevon was the third person involved with the project, after Walden and McQuillan, on loan part-time from the corporate accounting staff; later Tricia Hanafin provided accounting support. There were a few others involved with BBN Information Management Corporation whose names currently escape us.

32. J. M. McQuillan and D. C. Walden, "Portable Software for Electronic Mail Makes it Hardware-Independent," *Electronics*, March 10, 1981, pp. 167–171.

33. For at least one of the IBM ports of InfoMail, RATPL1 was used. The InfoMail team (primarily Ken Turkewitz) retargeted RATFOR to PL/1

34. Morris Keesan remembers that relatively late in the InfoMail business, he ported InfoMail "to a not-quite-UNIX operating systems called Unos that ran on an M68000-based machine made by Charles River Data Systems. It was part of a larger system that involved automated retrieval of stored mail from 3M 'WhisperWriter' portable terminals."

35. J. M. McQuillan and D. C. Walden, "Designing Electronic Mail Systems That People Will Use," *SIGOA Newsletter*, May l980, vol. 1, no. 2; *InfoMail User Guide*, BBN Information Management Corporation, Cambridge, MA.

36. As with the Hermes project, the InfoMail people heard repeatedly that no executive would ever do *his* own typing.

Chapter 20

SIMNET: A Revolution in Distributed Team Training

Compiled by David Walden

This chapter, compiled with the help of many participants, describes BBN's involvement in the development of SIMNET, an important innovation in training systems. The SIMNET chapter could have gone in Part III as it is arguably an application of computer technology or, more specifically, training technology (much other training technology is described in Chapter 13). However, we choose to categorize it with distributed systems such as those Rick Schantz describes in Chapter 18 and the networked e-mail described in the Chapter 19.

From early 1983 until the spring of 1993, BBN participated in the development of technology that revolutionized military training: the networked-simulator technology known as SIMNET as well as follow-on military training developments and procurements. The history and function of SIMNET is well documented.[1,2,3,4,5] This chapter sketches the SIMNET technology and its impact, primarily recounting BBN's participation in the project.[6] As typically happens when big dollars and the transformation of an industry are at stake, the story includes interpersonal and intra- and interinstitutional conflict.

20.1 Sketch of the basic technology

The definitive summary of SIMNET is the paper by Duncan Miller and Jack Thorpe.[1] This section abstracts some of that paper's content.

Simulation for training began to be widely used in the 1970s. The commercial and military pilot training systems are good examples. Trainers were built to aid instructors in teaching pilots how to fly a specific type of airplane. Because these trainers were substitutes for time spent flying in a plane (and because a human instructor or computer script could induce extraordinary conditions with which the pilot-in-training had to deal), these systems had to provide great fidelity to certain aspects of actually flying a plane and thus were very expensive. Such "substitution systems" also were available for training crews to operate tanks such as the U.S. Army's 70-ton M1 Abrams tank.

In the late 1970s, Air Force officer Dr. Jack Thorpe and others began to discuss an alternative use of simulation technology — not for substitution for the real plane or tank, but for allowing teams of "players" who already were expert users of their planes and tanks to practice in multivehicle situations (for example, combat) that could not be provided even in real vehicles. For instance, there was no economic or safe way except via simulators for the four-person crews of dozens of tanks and aircraft to work together practicing trying to overcome an opposing force of many other tanks and aircraft.

In the early 1980s microcomputer, wide-area-networking, local-area-networking, and computer image-generation technology was sufficiently developed to allow a major experiment in developing and trying a simulation system involving a large number of vehicles and players. This system was called SIMNET (SIMulator NETworking). The experiment was funded by ARPA (where Jack Thorpe was on assignment by that time) with expert support and cofunding from the U.S. Army. As described in the next section, SIMNET was developed by BBN, Perceptronics, Delta Graphics, and others, under Thorpe's leadership.[7]

A SIMNET tank simulator included a relatively inexpensive mockup of the inside of a tank cockpit and driver compartment with controls to steer the tank, fire its gun, and so on. Each simulator was designed on the principle of "selective fidelity," first articulated by Dr. Bob Jacobs (then of Perceptronics). That is, based on analysis of what the simulation was expected to accomplish, minimum fidelity levels could be identified for each component of the simulator. Some real-vehicle characteristics had high-fidelity simulations, some had moderate fidelity, some were abstracted, and some were left out entirely.

A real four-person tank crew resided in this tank mockup and operated the controls. In place of vehicle windows to the outside, the simulated vehicle had computer image-generation screens that showed views of the outside world as they would have been seen through actual windows. Each simulated vehicle was connected to a digital communications network. Computers calculated what a tank would do in response to movement of various controls and moved the vehicle in a virtual environment, showing the changed views out the windows. The simulated vehicles also had radio communication with other vehicles, command centers, and so on, which were digitized and conveyed over the digital network. The simulator computers included detailed topographical databases of the terrain the vehicle crews were practicing on; for instance, Fulda Gap terrain at the border of West Germany and the sphere of Soviet influence during the Cold War. Thus, the simulated tank could react appropriately as the driver steered it up a steep hill, through mud, or in other conditions; also, the simulated radio communication could involve line-of-sight and other radio propagation issues.

The communications network included local-area communications, so many simulators in the same room could participate in a training exercise, and wide-area communications, so simulators at different geographic sites could participate in a training exercise. Because of communications capacity issues, each vehicle simulator generated the graphical images out its windows from the motion of both itself and other vehicles on the network. When something changed (e.g., circumstances resulting from operator control of the simulated vehicle or from something done to one vehicle by another vehicle or by the environment), each simulator broadcast over the simulator network its location, its vehicle type, what it was doing (e.g., making a turn or firing a gun at a location, or burning up), and so on; and each simulator used this information to construct the "worldview" it showed to the crew of the simulated vehicle. In the absence of change information from other simulators, a particular simulator extrapolated the motion it displayed for other vehicles using the prior information it had about the direction and speed of the vehicle.

Using this technology, the crews of the simulated vehicles could work together and in opposition on the simulated battlefield, creating their own realistic human-motivated and driven battle, and a battle script was not needed. Depending on the topographical data available, military crews could practice anywhere on earth, including on an enemy's own terrain. These were revolutions in military training.

SIMNET included simulators for tanks (e.g., the M1), helicopters (e.g., the AH-64), fixed-wing vehicles, command posts, a semiautomated opposing force (SAF) capability, and pseudo-vehicles (e.g., the "flying carpet") that could observe and collect data on the simulated battle.

20.2 Evolution of the SIMNET project and BBN's participation

BBN people began to hear about the possibility of a networked simulation and training system in early 1982. On April 8, 1982, Craig Fields of ARPA phoned BBN division director Ray Nickerson to tell Ray that he was contracting with a company called Cinematronics to develop inexpensive ($1,500 to $3,000) video training systems, to be applied to the training of tank gunners and other similar problems. Nickerson's memo said,

> [Fields] wants to . . . find a contractor who can take delivery on several hundred of these devices, hook them together in a network, and develop the software that will make it possible to conduct war games on them. As I understand it, the general idea is to permit a large number of people, each with his own terminal, to participate in the same simulation. This means developing both network software and courseware. Craig describes the network software as the "trivial" part of the problem.

Fields also told Nickerson that the ARPA program manager for the project was Jack Thorpe.

Nickerson was leading BBN's Division 4, which was where the BBN education and training group resided. People from BBN's Division 6 (Frank Heart's division) also heard about this prospective program and began to talk to the people at ARPA. Such interdivisional competition was not uncommon. Jim Calvin (Division 4) remembers,

> [The] idea [was] to use the technology found in electronic games of that era to create a new class of training device for the DoD. The system was to be inexpensive, possibly networked, and allow free play. . . . BBN was asked to look at the idea. . . . At the beginning of the effort, folks in Frank Heart's division were looking at the problem from a network point of view, involving particularly Rob Gurwitz.

In December 1982, Nickerson was again talking on the phone to Craig Fields and heard that BBN people were meeting with Fields that afternoon about the project. Fields was surprised to learn that Nickerson thought Division 4 was out of it, and Fields made clear he had other intentions.

Dan Massey remembers,

> BBN submitted two proposals, one from Divison 4 and one from Division 6. DARPA selected BBN and Perceptronics to perform the initial work, but wanted the ideas from both BBN proposals included in the effort. To share responsibility for the project, Division 4 got to appoint Duke [Miller][8] as program manager and to place the contract in Dept 43 [the education and training department], which Duke had taken over from Wally Feurzeig, who, of course, remained involved. Division 6 in return got to name the chief architect or lead designer . . . for the system. [In time,] he [perhaps Rob Gurwitz] left BBN . . . , effectively ceding control to Duke and Division 4.[9]

Jim Calvin remembers the system development.

> The decision was made to simulate tanks rather than fighter aircraft. There were several reasons for this: tanks moved more slowly so the simulation task would be easier (this turned out to not be true), but more importantly, it would avoid direct comparisons between these inexpensive systems and the $10–20 million systems used by the Air Force for their trainers.
>
> Early in the program, members of the BBN team visited Fort Knox to see, drive, and fire the M1 tank. This was quite an adventure to behold. We were given various orientations to how the Army organizes, trains, the doctrine for various maneuvers, etc. One vivid memory is that of a staff sergeant informing us that ". . . an M1 was a

Figure 20.1. BBN people in front of a tank at the Fort Knox museum the day after they drove and fired the tanks. From left to right are Duncan Miller, Dave Epstein, Maureen Saffi, Dick Koolish (below), Phil Yoo (above), Joe Marks (below), Jim Rayson (above), Jerry Burchfiel, and Jerry's son just behind him. (Photo courtesy of Jim Calvin, who took the photo and thus is not in it.)

killing machine, and it doesn't care who it kills. So when I tell you to pay attention, you pay attention." We did. We were permitted to load the M1 with range ammo. We fired at targets on a range, hitting most of the targets — and one 55 gallon drum that wasn't actually a target. Joe Marks, at the time an Irish national who had never driven a car, was able to drive an M1. The noise from the main gun was deafening and it seemed like the entire 61 ton machine jumped when that gun went off. All of this occurred while enlisted and officers alike watched in disbelief as a motley crew from Cambridge, Mass., was afforded unfettered access to these systems. However, many detailed questions were answered including "does pivot mode really pivot around a point, or does the tank walk in some direction while pivoting?"

Originally BBN people believed they might do the entire job, networking, platform (tanks, jets, etc.) simulation, and computer generated graphics. There were sets of people considering each of these components. Duncan Miller, Jerry Burchfiel and others worried about the simulation of a mechanical platform in motion and how the protocols would manifest various aspects of interactions between distributed systems. The data required to represent platforms in motion and the interactions between the platforms (collisions, shots being fired, rounds striking objects, etc.) were then turned into network packet descriptions.

Generating graphic images for the system appeared to be a difficult problem, especially for the original target price ($10–25K for the entire system). Ray Tomlinson, Jerry Burchfiel, I, and others developed numerous techniques that simplified the scope of computation required to generate an image. Remember at this point in time, a 16MHz 68000 was a pretty hot chip. The sponsor [ARPA] viewed some computational simulations of the graphics that this inexpensive approach would produce and was unimpressed.

Shortly after the graphics demonstration, Thorpe introduced BBN to a group at Boeing that was heavily involved in high performance graphics systems. They used a more classical approach used in aircraft simulation systems. The images were textured and provided better depth perception than those shown earlier at BBN. It turned out that Boeing was not interested in following this endeavor, so the staff involved left Boeing to form Delta Graphics [in Bellevue, Washington].[10] With this, the graphics effort left BBN. This increased graphics fidelity came at a price; our target system price was now $100,000.[11]

Meanwhile Thorpe wanted to improve the fidelity of the training device by creating an environment that looked, felt, and sounded like a tank. He found a group in the LA area called Perceptronics that specialized in such areas. Perceptronics ended up responsible for the shell and all the controls that comprised the simulator.[12]

This final decision by Thorpe created the interface boundaries for the system. Delta Graphics was responsible for the computer generated, out-the-window views.[13] Perceptronics was responsible for the look, feel, and selection of all of the control, displays, sounds, etc., that made up the training device. BBN was responsible for the vehicle simulation, networking, interface to the Perceptronics controls and Delta Graphics visuals, and the integration of the system. BBN would also end up with the responsibility to develop all of the adjunct systems that provided logistics, support, indirect fire, etc. to support the training systems.

The decision by Thorpe to move the graphics and training device to other contractors also created some problems at BBN. Some members of the team left at that time (I believe this was around the time when Gurwitz left as technical lead). But by March 1984, the team was pretty well in place. At that time Duncan Miller was the program manager for BBN. Jim Calvin was the technical lead. The technical team included at least the following people: Ed Burke, Duncan Miller, Jim Calvin, Jim Rayson, Dick Koolish, Dave Epstein,[14] A. Chatterjee, Phil Yoo, Joe Marks, Maureen Saffi, Michael Harris, Rob Gurwitz, Will Crowther, and Jerry Burchfiel. Some of these team members left SIMNET or BBN, and many others eventually joined BBN's SIMNET team.

By early the project, an architectural approach was fixed in everyone's mind, but the detailed designs were yet to come. While BBN, Delta, Perceptronics, and DARPA all believed the system could be built, the rest of the world was much more skeptical. As we visited various groups already doing simulation and/or training and described the way the SIMNET system would work, the majority of people told us we were nuts, the thing would never work. But back then, almost nothing was networked, and microprocessors were toys that couldn't do very much — certainly nothing useful for a military simulation or trainer.

Early in the project, Thorpe had found some expert advisors to help him with the program — Gary Bloedorn, Neal Cosby, and Ulf Helgeson. Col. Gary Bloedorn, who had recently retired as the director of training development at Fort Knox, was chiefly responsible for establishing the ties to the Army that enabled Thorpe to raise much money for the experiment and keep it going to a full transition.[15] Neal Cosby had recently retired as the Army colonel who had headed the Army Research Institute. Of Cosby and Helgeson, Duncan Miller says,

[Cosby's] primary contribution to SIMNET in the early years was his vast network of contacts. If he didn't know the right person who could get something done, he knew who would know.

[Ulf Helgeson of Perceptronics was an] industrial designer who was able to develop realistic controls and displays at low cost. He also conceptualized and designed the "December demo" layout, leading visitors (many of whom had very little exposure to computer graphics, networking technology, etc.) through logical and effective demonstrations of some rather advanced concepts.

Perhaps the key to the SIMNET program's success was that Thorpe managed to secure sufficient funding. Writing in 2003, Duncan Miller explained this:

> Thorpe was never able to obtain any support from his own service, the U.S. Air Force. Only in the last few years (since the late 1990s) has the Air Force committed to a Distributed Mission Training (DMT) program. That's nearly twenty years after Thorpe began presenting his vision, only to be rebuffed by leaders who saw only a threat to the number of actual flying hours per year.
>
> It was the U.S. Army that stepped up. . . . According to what Jack Thorpe told me, a critical step occurred when MG Frederick Brown, then Commandant of the U.S. Army Armor School at Fort Knox, KY, happened to hear Thorpe make a presentation about his ideas. This struck a resonance with Brown, because his recently retired Director of Training Development, Col. Gary Bloedorn, had been working to find ways to link the Army's new, multi-million-dollar M-1 tank Conduct of Fire Trainer (COFT) simulators together via a local network. This effort had foundered because the experts said it was impractical, as well as extremely expensive. Listening to Thorpe, Brown recognized that what Bloedorn had been talking about might be achievable after all. He put Thorpe in touch with Bloedorn and offered them broad access to the Armor School's facilities and expertise. . . . young SIMNET developers received hands-on familiarization with M-1 tank operations and tactics from the Armor School's best instructors.[16]
>
> As initial SIMNET development proceeded, Thorpe and Bloedorn began briefing senior Army leaders about a vision of several hundred networked simulators—enough that entire brigades could participate in joint training exercises. However, it was obvious that even if Thorpe's ambitious cost targets could be achieved, many tens of millions of dollars would be required. The Army had no such programs budgeted, and the process of approving new programs typically took years. DARPA represented a way to channel funds outside the normal process—but the money had to be found, and the concept had to be approved at the highest levels of the Army leadership.
>
> To build support for this challenge, Thorpe arranged for a concept demonstration, showing mockups of low-cost simulators, graphic images recorded on videodiscs, and a crude demonstration of real-time simulation between simulator mockups using controls and instruments mounted on relay racks. The effect of this "December demo"—which actually occurred during January and February of 1985 near DARPA in Rosslyn, VA—was electrifying. By the time it concluded, more than 100 general officers (plus many more of their senior staff) had experienced the demonstration. This number included at least half of the Army's 4-star generals. Many were astonished by what they experienced, the result not only of the technology they saw being demonstrated live, but also by the fact that they were being briefed at every turn by young engineers and computer scientists in their mid-twenties. In their experience, serious program briefings were usually given by senior managers in their forties or beyond. Part of the effect was due to the fact that the young engineers were completely unfazed by briefing the most senior officers. This became apparent to me when two of the youngest developers asked me what the stars and eagles mean on the officers' uniforms, and which of these were the higher ranks![17]
>
> After all the publicity generated by this unprecedented event, two senior Army leaders agreed to champion the development of SIMNET and the installation of an initial suite of some 250 simulators to be spread across multiple training sites.
>
> One of these champions was Gen. Max Thurman, Vice Chief of Staff of the Army. He envisioned how SIMNET could revolutionize the training of Army combat teams, giving every division the means of training battalion-on-battalion forces

> in simulation, greatly extending the impact of the Army's relatively new National Training Center (NTC) at Fort Irwin, CA, where Army battalions could practice realistic warfare in the desert against experienced Opposing Forces (OPFOR), who were highly proficient in fighting as (and pretending to be) Soviet armored regiments. After rehearsals using SIMNET on a realistic representation of NTC terrain, vs. opposing forces, armored units would arrive at the NTC with a much higher state of readiness, saving days of initial orientation and practicing of basic maneuver techniques and allowing them to focus on the advanced training they were there for.
>
> Another champion emerged from an unlikely source — the Honorable James Ambrose, Undersecretary of the Army. I will never forget meeting him in his massive office in the Pentagon. He was an elderly, unassuming grandfather in a rumpled cardigan sweater. He sat at his desk, eating a peanut butter sandwich and drinking from a carton of chocolate milk, while he talked with Jack Thorpe and me (only the three of us were present). After asking us many questions, he said that he was prepared to see that sufficient funds were found and diverted from existing Army programs to ensure that SIMNET was developed and put in place. His reasoning was astonishingly far-sighted, even beyond Thurman's. He envisioned that, if we succeeded, it would fundamentally change the way the Army developed and acquired weapons and sensor systems. Proposed concepts could be implemented first in simulation and used by the troops who were the ultimate users of the new systems. Real users could work through realistic scenarios using equipment that was not yet built, developing tactical procedures and uncovering potential shortcomings and problems before billions of dollars had been spent (the ill-fated field test of the "Sgt. York" division air defense gun was still painfully fresh in everyone's minds). In articulating this vision, Ambrose anticipated by many years the Army's "Simulation-Based Acquisition" initiative. Through the late 1980s, Developmental SIMNET (SIMNET-D) facilities were built at the Armor School at Fort Knox, KY and the Army Aviation (helicopter) School at Fort Rucker, AL. At these facilities, and later others, early evaluations were conducted of several proposed systems, including digital communications and in-vehicle graphics to provide command, control, and situation displays. These prototypes evolved rapidly into the highly capable systems being used now in Iraq.
>
> Many budgetary battles were fought behind the scenes, very few of which ever became known to the development team working on rapidly implementing new SIMNET capabilities. But undoubtedly, without the active support of Thurman and Ambrose, SIMNET would probably have remained at the "science-fair" demonstration stage.

In parallel with the external organizational issues, there were also BBN internal organizational issues. With an eye to better exploiting the promising SIMNET technology, BBN Systems and Technologies president Dave Walden eventually set up the SIMNET project as a separate division (BBN Advance Simulation) with Al Stevens[18] as division director and Duncan Miller as his key deputy.

The various participants — Thorpe's advisors, BBN, Delta Graphics, and Perceptronics — settled into a reasonable working relationship and the project moved ahead well. However, in time both BBN and Perceptronics bid to buy Delta Graphics, as BBN and probably Perceptronics each saw the possibility of becoming the dominant partner in the SIMNET program and setting itself up for future training procurements based on using the SIMNET approach. BBN won the bidding and bought Delta Graphics in 1987 for approximately $16 million.[19] The Delta Graphics people were willing to sell because they were facing the need to acquire additional capital to support R&D and manufacturing as the company grew. Former Boeing engineers Mike Cyrus and Drew Johnston had been clever enough to fund their start-up without seeking outside funding (this is possible when your main enterprise is a government contract with progress payments);

seeking venture capital funding would have required dilution of the Delta Graphics founders' stake in the company. Selling the company to BBN meant the founders themselves collected the full then-market value of their company. They also preferred selling to BBN rather than to Perceptronics because they saw a better cultural match with BBN. Delta Graphics became a division of BBN Systems and Technologies, with Mike Cyrus reporting to Dave Walden in parallel with Al Stevens and the Advanced Simulation division.

Lots of effort went into the collaboration between the two divisions. There were interchanges of staff members between the Delta Graphics people in Bellevue, Washington, and the Advanced Simulation people in Cambridge, Massachusetts. Mike Cyrus recommended that Touraj Assefi from Boeing be hired to live and work at BBN in Cambridge (in Frank Heart's division) for a couple of years so he would know the company well and could later return to Bellevue to help with the management of the office there. BBN's manufacturing people (e.g., Gerry Davidson) got involved with manufacturing the computer image-generation devices of Delta Graphics.[20]

All too soon, however, Cyrus and Johnston announced they were resigning to do other things in life (as so often happens after a company is bought from its founders). While Delta Graphic's manufacturing manager, Dave Bell, took over temporary responsibility for the group in Bellevue, people from Cambridge — including Walden himself and one of Stevens's staff, Dave Johnston — also spent lots of time in Bellevue working to hold the group together. Soon the Bellevue division was merged into Stevens's division with Bell reporting to Stevens. Eventually Assefi was offered Bell's job and returned to Bellevue; and Dave Bell, who was to come to Cambridge to help Stevens, instead left the company. There were continuing challenges of managing this large, distributed combined group in the highly visible SIMNET environment, and eventually BBN president Mike LaVigna sent Gerry Davidson, a highly experienced and capable general manager from elsewhere in BBN, to help manage the Advanced Simulation division.

20.3 Key ideas and challenges

On the networking and simulation side of things, Jim Calvin says,

> The concept of using dead reckoning models to reduce network traffic was one of the most important techniques developed in SIMNET. The notion is that each simulation runs a low fidelity model of every other simulated entity in the system. Each local simulation runs a high fidelity and the low fidelity model. When the high fidelity model deviates from the low fidelity model by an agreed upon threshold, a new update is generated for all observing simulators to use. So rather than generating position, orientation, etc. updates every 1/15 of a second (or whatever the simulation rate might be), updates might only occur once every 10 seconds (the time-out threshold). Use of this "dead reckoning" concept allowed us to do two things: first, keep up with the updates. In those days, computers and network interfaces were pretty wimpy. With more than a handful of simulators on the network, the simulation processors would have been totally consumed just processing updates. The second thing was closely related, it allowed exercises to scale to large endeavors of up to 1,000 entities.
>
> This dead reckoning approach also became the basis for many network based games. At a much later time I attended a virtual reality conference that basically gave credit to SIMNET for inventing most of what they needed to do multi-player virtual reality systems.
>
> This scaling of entities would be an ongoing area of research, including a 1991–1992 BBN program called 10^4, or how do you handle 10,000 entities. Dan Van Hook,

> Jim Calvin and others worked on this project and the concepts developed during the project have been used repeatedly in the years since.
>
> The early technical challenges were primarily focused on finding clever ways to reduce processing demands and to facilitate exchanges of information between the system components provided by the three contractors. The most challenging of course was the exchange of information between the simulation software and the image generator. There were countless discussions and arguments about which side should be responsible for what. Eventually a manageable agreement was reached about what data would be passed, and when in the image generator's update cycle it would occur. This exchange timing was critical as it strongly affected the latency between the creating of the data in the simulation and a visual scene representing that data.
>
> Discussions continued for several years (but were definitely easier after the purchase of Delta by BBN) regarding the database that the image generator (CIG) used for its scenery. The semi-automated forces (SAF) needed a similar, but much less rich version of the database (the visual portion of the data was not needed). Eventually tools were refined so that the CIG and SAF databases could be directly generated from a single source.
>
> One of the first problems on the simulation system side was to find a computing platform and operating system that could provide real-time capabilities. After a long search, we ended up using the Masscomp system, which was UNIX based.
>
> On the network side, handling the load of simulation traffic was an ever present battle. One of the techniques that we ended up using was to use programmable network interfaces that allowed us to move much of the network processing from the main processor to the network card. It's interesting to note that some high performance network interface cards (NIC) these days essentially do the same thing by moving some or all of the TCP stack into the NIC.
>
> The interface to the control system was also engineered to reduce the load on the main processor. Special boards . . . with a separate micro-processor were developed that had a single serial interface to the main processor. Each [board] had a large set of input and output lines for driving lamps (LEDs), sensing switches, and A to D and D to A ports. The micro-processor on the IDC constantly polled the various inputs and reported any changes to the main processor. Another interesting cross-over, the first early IDC prototype was a re-worked Jericho prototype serial interface board with modified code.

On the computer image-generation side of things, Drew Johnston explained the following:

> DARPA was trying to solve a networking problem with SIMNET, but they needed thousands of these simulators and the cost of the graphics component was a blocking factor without a significant reduction in cost. The Computer Image Generation (CIG) system [from] Delta Graphics provided [the necessary cost] breakthrough with an 8 channel, texture mapped, anti-aliased, real-time visual system combined with an integrated real-time database traversal engine [at 1.5 percent] of the cost of traditional simulators.
>
> In the early 1980s visual simulators cost between $1M and $4M per channel. The primary breakthrough for the SIMNET computer image generator (CIG) was that we were able to provide an 8 channel visual simulator for around $120,000. . . . The closest alternative at the time was the new Geometry Engine provided by Jim Clark's newly formed Silicon Graphics. This was investigated by DARPA and at the then cost of around $60K per channel that would still have cost a minimum of $480,000 just for the equipment. The Delta Graphics CIG was still [one-fourth the cost] and was provided in a single chassis instead of eight high performance workstations that would need to be synchronized to provide the eight visual channels. In addition

there were other technical issues related to texture mapping and anti-aliasing that also couldn't be resolved by using the Geometry Engine at that time.

The primary technical breakthrough at the time was the use of video RAM using a depth buffer to resolve the occlusion problem. The early 1980s marked a point where the cost of RAM was starting to decline to the point where large video RAM buffers could be used. Until this time it was too expensive and too large to be considered feasible for this kind of application. Leading up to the early 1980s, RAM in the low megabytes still cost in the order of tens of thousands of dollars in a form factor that filled boards [of] several feet square.

The added benefit of exploiting the depth-buffer architecture was that terrain databases could be built much more quickly since it didn't have the problems associated with separating planes and other approaches (BSP, etc.) to solving the hidden surface problem. Delta Graphics was involved in leading edge research in rapid database deployment for DARPA that could deliver databases of real world locations for use in tactical training and mission rehearsal in a couple of days. The continual goal was to drive this down to a matter of hours.

The depth buffer and texture mapping have become pervasive now, but in the late 1970s and early 1980s texture mapping and depth buffer algorithms were just starting to show up in research papers of the time. The SIMNET CIG was the first real-time computer image generator that exploited this architecture. These features are now commonplace on video cards for PCs and can be seen in most common video games.

20.4 Demonstrations and applications of SIMNET technology

Between 1987 and 1989, there were a number of prototypes and experiments with various elements of SIMNET, and the system was made operational in January 1990. In their paper, Duncan Miller and Jack Thorpe say, "[I]t is not surprising that the large majority of SIMNET applications have been training exercises. These are conducted at battalion level (80 manned combat vehicles, plus 200–300 [semiautomated forces] entities) at two sites: Fort Knox, KY, and Grafenwoehr, Germany. The other sites run company-level exercises (up to 20 manned combat vehicles, plus 50–100 SAF entities)." They also mention several "cooperative developments and exercises with other DoD organizations aimed at evaluation of hypothetical weapons systems, tactics development, and rehearsal for field test and evaluations exercises."[21] Specifically:

- In two exercises in 1988 and 1989 at Fort Bliss, Texas, where the Army Air Defense Artillery School rehearsed field tests for the Forward Air Defense System.
- In 1988 at Fort Knox, the Army Research Institute studied combat vehicle command and control.
- In 1989 and 1990 the U.S. Army Missile Command sponsored a rapid-turnaround simulation of a proposed non-line-of-sight missile.
- From 1990 to 1991 a DARPA-sponsored consortium assessed the effect of laser weapons on the battlefield.
- In 1990 the U.S. Army Missile Command sponsored a study of a weapons system that was under development.
- After the 1991 Gulf War, SIMNET was used to recreate shot-by-shot a key armor/cavalry battle — "73 Easting"[2]

As ARPA, BBN, Perceptronics, and others continued to advance the SIMNET technology, the need became apparent for a secure, wide-area network to support the IP-based stream protocols used in SIMNET, as well as other stream protocols used for video teleconferencing, which were incompatible with the TCP/IP based Internet. In response to this need, the Defense Systems Information Agency developed and deployed what was initially called the Defense Simulation Internet (DSIN). The DSIN is now called the Defense Research and Engineering Network (DREN). In Steve Blumenthal's assessment, "The Defense Simulation Internet and SIMNET were major systems activities that improved military exercises via distributed training and contributed to the U.S. success in Desert Storm." Assefi adds that for Operation Desert Storm, "It was very easy to simulate Iraq because of the geography; we did have a desert storm scenario that was used."

Important aspects of the SIMNET that BBN helped develop were adopted as part of the Distributed Interactive Simulation (DIS) standard (IEEE Standard 1278-1993).

20.5 BBN's attempt to expand into the military training business

In time, the SIMNET technology attracted the attention of both U.S. military training people and their counterparts in other countries (particularly Germany). Also, ARPA was feeling it was time to transition the SIMNET activities out of ARPA and to the military services. Several big training procurements looked ripe for use of SIMNET-like technology. Thus, BBN's Advanced Simulation Division (as the SIMNET and related activities were now called) began to pursue some of these future possibilities.

In Germany there was a competition to build a new training capability known at AGPT (Aus Gehfechtssimulator vor Panzer Troopen). Chris Harz of Perceptronics (which was typically a step ahead of BBN in promoting itself and getting to know prospective customers for the SIMNET technology) sold the Wegmann GmbH company of Kassel Germany (a maker of parts of Germany tanks) on bidding the SIMNET technology on AGPT. When Al Stevens and his marketing person, JC Williams, heard about this opportunity, they lobbied with Olaf Escheler and Wolfgang Kratzenberg of Wegmann for BBN to participate in parallel with Perceptronics in Wegmann's AGPT bid, arguing that BBN had the key SIMNET technologies for simulation, graphics, and networking. The Stevens-Williams lobbying effort succeeded, and one day in late February 1989 BBN Systems and Technologies president Dave Walden received a phone call from Wegmann's president pleading for BBN to *immediately* send a team to negotiate a teaming agreement to make a bid to the German government. The BBN technical and negotiating team[22] flew that evening to Brussels. In the early morning a private jet sent by Wegmann took them to Frankfurt, and from there they were chauffeured at *high speed* on the rainy German autoban to Kassel for a rich lunch (and drinks) hosted by company owner Manfred Bode and company president Werner Zimni; then they were taken to the factory to begin an all-afternoon-and-evening design and negotiation session. Part of the reason for the inclusion of the high-level participants from each company was to build a relationship for bidding on several German training contracts, which therefore involved lots more wonderful eating and drinking together during this visit.

A team consisting of Dan Massey, JC Williams, Jim Panagos, and others went to Germany to help Wegmann finalize the AGPT proposal (graphics system experts Rick Bess and Mark Kenworthy also were in Germany several times as the bid was being prepared). Wegmann was the underdog in this competition; the leading German training company was Atlas Elektronik. Wegmann was arguing that its bid should get the contract because it brought the SIMNET technology. However, the German government doubted the U.S. government would allow such technology knowhow to be exported. Sometime after the bids were in, during the source selection phase of the procurement, Jim Shiflett, then SIMNET program manager in ARPA, visited Germany with members of the BBN team and stated that the technology knowhow could be exported. The German government procurement people then began to favor the Wegmann-BBN bid; they were particularly impressed with the number of moving models BBN's image-generation system could display with decent fidelity (as described to them by Rick Bess of BBN's Delta Graphics division). In the end, Wegmann was awarded a 220 million DM (about $120 million at the time) contract of which BBN's part was $65 million. JC Williams says, "This number will be burned in my memory forever . . . there is a picture taken on Valentine's Day in our conference room at 33 Moulton Street of Wolfgang Kratzenberg and us with a heart around the $65 million number signifying final negotiations and a sweetheart deal."

While the AGPT system was based on the SIMNET technology, it was rewritten in Ada and some changes were made (e.g., Jim Calvin remembers that it did not use dead reckoning for updates). Dan Massey led this reimplementation effort using a team of people who were not part of the original SIMNET team.[23] In the end, about 49 simulators and 16 platoon sets were delivered by BBN; both Massey and Williams suggest that AGPT was one of the most profitable contracts in BBN's history (something like $20 million in profit).

Later, Wegmann and BBN also bid on the German Marder training system. However, having lost the AGPT bid, Atlas Elektronik saw a major threat to its business and decided it would not lose again. For the Marder contract, Atlas Elektronik underbid the Wegmann team and also made disparaging remarks about the SIMNET technology's capabilities. This time the Wegmann team lost.[24]

Back in the United States, BBN had a small office in Orlando, Florida, near PM Trade (Program Management for Training Devices), the Army training procurement center, which was gearing up to do a major procurement known as CCTT (Close Combat Tactical Trainer). For this procurement BBN teamed with Martin Marietta (which had the role of prime contractor), Perceptronics teamed with Loral (as prime contractor), and teams led by IBM's Federal Systems division and GE's training division also bid. JC Williams believes that the Martin Marietta/BBN team had the contract won, having bid $380 million. However, the procurement was so large that PM Trade could not do the source selection itself; that function had to go higher in the Army chain of command. The CEOs from each bidder's prime contractor were called to visit Mike Stone who said he didn't want any buying-in and sent the CEOs back home to make a final conservative bid. Norm Augustine of Martin Marietta asked his team and BBN to come up with a conservative bid, which they did: $480 million. Presumably the Loral and GE teams also followed Mike Stone's admonition to submit conservative bids. However, IBM didn't blink and stuck with their original bid of $409 million and won the procurement. JC Williams believes IBM's bid was less than half of its final costs. Duncan Miller believes that, "the Army was not willing to bet on an upstart graphics system. They preferred the 'deep pockets' approach of the prime contractor, IBM Federal Systems, who [was] willing to bid the highest-risk element, the unprecedented development of 'accredited' Semi-Automated Force components, on a fixed-price basis." Miller also heard from good sources that "IBM lost huge sums of money."

BBN had also opened an office near the Army's tank training center at Fort Knox, Kentucky, and was continuing to support the original SIMNET work there and elsewhere.[25] In time, however, the SIMNET work was recompeted via the Army Simulation Training and Instrumentation Command (STRICOM, a renaming of PM Trade), and BBN lost the SIMNET T (Training support) contact to IBM Federal Systems and lost the SIMNET ADST

(Advanced Distributed Simulation and Training R&D) contract to Loral. Duncan Miller believes that Loral bid less than half of what BBN bid for the ADST contract, expecting to play a change order game to gain major contract expansions over their initial bid. However, according to Miller, it didn't turn out this way: The Army spent elsewhere the rest of the money it had expected to spend on ADST, and "for the next few years, only the most basic and routine activities took place in what were previously the SIMNET facilities, sadly squandering their immense potential." BBN did continue to have some task order contracts directed to it through the Loral contract.

Next BBN lost a bid, known as WarBreaker, from its old customer, ARPA. Dan Massey explained,

> Thorpe initially said it was a "SIMNET follow-on" (it came out before ADST, which was closer to being the SIMNET follow-on) and urged BBN to bid. Nothing could have been further from the facts. DARPA issued a 20 page statement of work and allowed a 10 page response. There was no way BBN could have won this job because nobody at BBN, in marketing, management, or technology, and none of the subcontractors we selected had the faintest idea what WarBreaker was about. I am pretty sure Thorpe didn't know either. The job was won by Booz Allen Hamilton teamed with SAIC. The actual subject of the procurement was so highly classified that not one single mention of it appeared in the RFP. You had to know what was wanted or you were out of luck. . . . First, the selected proposal team assembled and wrote their 10 page response (which was total junk, in retrospect). Then Duke . . . [reviewed] it and saw it didn't connect to SIMNET in any way, and we (he and I) sat down to rewrite it. The result made Duke happy, but, because he didn't know what he didn't know about this job, was actually totally irrelevant. . . . The government person in charge of the bidder debrief said, rather uninformatively, that we didn't have enough "systems engineering." This was one of a hundred equally significant things we didn't have about WarBreaker. . . . [This led division management to worry about] software engineering, requirements definition, and all the things that SIMNET (and much of BBN) had never really thought much about doing in a formal way.

Later Massey learned that the WarBreaker procurement was for a completely different purpose and had nothing to do with SIMNET. Thus, both sides of the internal BBN I-know-better-no-I know-better debate were wrong.

Jim Calvin remembers that "The cumulative effect [of losing these bids] was making it difficult to sustain the division. Quite a few staff members were moved to other BBN divisions that needed help at the time, for instance, the speech group."

Eventually, because of other financial problems in several parts of BBN and the poor showing the simulation division had made on winning contracts, and despite BBN's substantial investment (including the purchase of Delta Graphics, which was justified in terms of the business BBN could win with its own computer image-generation activity), BBN president Steve Levy, who had met Loral's president Bernard Schwarta, sold the Advanced Simulation Division to Loral in 1992 (the sale was consummated in 1993).[26] Thus, Loral got to keep the task order business it had been sending to BBN under the SIMNET ADST contract and got the SIMNET technology, including the Delta Graphics computer image-generation capability.[27] Al Stevens, JC Williams, and Jim Calvin, among others, went with the business to Loral. Miller joined another part of BBN and did not go with the business to Loral.

Other parts of Loral's training business were consolidated into the until-recently-BBN division, and this entire Loral activity operated out of BBN's 50 Moulton Street building, which Jim Calvin remembers created "interesting problems."

20.6 Non-BBN spin-offs and follow-ons of the SIMNET activity

The next decade (1993 to 2003, when this was written) saw various non-BBN spinn-offs and follow-ons based on the SIMNET effort. The story did not end in 2003, of course, but this section doesn't go beyond then.

The sale of the advanced simulation business prevented BBN from working in the modeling and simulation area for five years; thus, Duncan Miller left BBN and in September 1993 created a new group at MIT Lincoln Laboratory, working in the same networked training area. Duncan Miller was invited by Jimmy Shiflett (Thorpe's successor at DARPA) to become technical director of the newly established Defense Modeling and Simulation Office (DMSO) in the Washington area. Miller declined to relocate, however, and the job went to someone else. Shiflett then funded Miller's group at Lincoln Laboratory. Jim Calvin joined Miller at Lincoln Laboratory, as did several other longtime participants in the SIMNET group (e.g., Carol Chiang and Dan Van Hook).

(Duncan Miller himself returned to BBN in 2001, first as a consultant and then as a full-time employee; where he assumed management of BBN's Mobile Networking and Systems Department. In 2008 Duncan retired from BBN, but he remained active on national and international committees related to training and simulation.)

Warren Katz and John Morrison, who were with the BBN SIMNET project from 1987 to 1990, founded MÄK "to provide cutting-edge R&D to the DoD in the areas of distributed interactive simulation (DIS) and networked virtual reality (VR) systems and to convert the results of this research into commercial products for the entertainment and industrial markets. MÄK's first commercial product, the VR-Link developer's toolkit, is the most widely used commercial DIS interface in the world."[2] Dan Massey notes, "MÄK has been a major force in educating the simulation community about distributed simulation and providing the basic tools for them to use, without having to reinvent it all for themselves. Their focus is increasingly commercial, rather than for defense."

Paul Metzger, who was with the BBN SIMNET project, and his wife founded Reality by Design (RBD). Metzger developed an interest in dimensional audio processing and experimented with coupling directionally coded audio to image generators. Several exBBN people went to RDB, which, like MÄK, was attempting to commercialize SIMNET technologies. RDB operated for several years before selling out to another company. Paul Metzger and Art Spear (ex-BBN and ex-RBD) now work at MIT Lincoln Laboratory, but now with real aircraft as well as simulations.

BBN SIMNET participant Stuart Rosen and Delta Graphics cofounder Drew Johnston created WizBang! Software Productions, Inc., which, using experience and ideas from the SIMNET world, provided 3-D environments for the games Hyperblade and Baseball. Later Microsoft acquired WizBang.[2]

Al Stevens and Steve Levy founded Kaon to develop an Internet-based distributed computer gaming system, although in time Kaon has evolved into other businesses.

Brian Soderberg also started a gaming company when he left BBN/Delta Graphics, and Rick Bess worked with Quantum3D (which provides interactive graphics technology) for a while.

One of the most interesting SIMNET spin-offs, according to Dan Massey, was MetaVR:

> Around the time of the sale to Loral, Garth Smith, who had only worked on AGPT (we recruited him as he was about to be laid off from BBN Software Products) left to go to TASC and then quickly left there to establish MetaVR, a distributed company (no single place of business, all networked), which designs and manufactures image generators based on commercial graphics cards (game cards) for the standard PC, as well as the software, communications products, and tools to enable them to

> replicate SIMNET and much more. Among their numerous marketing successes, MetaVR eventually won contracts to replace essentially all BBN image generators the government had purchased. This was relatively easy to sell since the MetaVR package now cost less than one month's maintenance on the replaced BBN equipment, while having greater capabilities (Moore's Law at work, of course . . .).

Also, training companies SAIC, Lockheed Martin, and others in the military training community all ended up with staff from the BBN SIMNET team. Dan Massey has reported the following.[28]

> Rob Calder Alan Evans, Ben Wise, and I left Loral's SIMNET group to form an SAIC office in Burlington, MA, where we were later joined by Rob Vrablik, Jim Panagos, and a few others. I set up and rather casually managed this office for two years until I moved to the DC area to work more closely with the STOW 97 team. Alan Evans took over the Burlington office from me. Anthony Courtemanche moved to Orlando for Loral, then joined the SAIC office there working on CCTT and ADST II.
>
> CCTT had been won by IBM Federal Systems Divsion, which was soon purchased by Loral (which had won SIMNET ADST and had bought BBN's Advanced Simulation Division) which was soon purchased by Lockheed, which had merged with Martin Marietta about the same time. The subsequent follow-on, ADST II (very original), was won by an SAIC/Martin Marietta team. IBM/Loral/Lockheed was teamed with SAIC to provide everything complicated in the system, thus establishing (with ADST II, STOW 97, JPSD, WARSIM, JSIMS, NASM, and J-SIGSIM to name a few) SAIC as the primary inheritor of the BBN Advanced Simulation legacy. (SAIC was/is the primary M&S technology supplier to all these programs, as well as FCS, OFW, TIA and other recently developed programs.)
>
> Pretty much every other BBN SIMNET developer [not mentioned above] eventually ended up under Greg Swick's management at Lockheed (in Burlington, MA). This was essentially all the remaining OPFOR developers, especially the more junior members of the team. The remainder of the BBN/Delta Graphics group in Bellevue, Washington, (led by Dale Miller), reports to the Lockheed branch in Burlington, MA.

In more recent years, the U.S. Air Force and Navy have been procuring "distributed mission training" systems, essentially modern incarnations of the SIMNET idea. Jack Thorpe's ideas, together with BBN and its people who helped Thorpe develop and implement those ideas, have revolutionized the training world.

Notes and References

1. Duncan C. Miller and Jack A. Thorpe, "SIMNET: The Advent of Simulator Networking," *Proceedings of the IEEE*, August 1995, pp. 1114–1123.

2. Tim Lenoir and Henry Lowood, "Theaters of War: The Military-Entertainment Complex," in Jan Lazardzig, Helmar Schramm, Ludger Schwarte, eds., *Kunstkammer, Laboratorium, B$^1/_2$hne-Schauplîtze des Wissens im 17. Jahrhundert/ Collection, Laboratory, Theater*, Berlin; Walter de Gruyter Publishers, 2003 in both German and in English; available at www.stanford.edu/dept/HPST/TimLenoir/Publications/Lenoir-Lowood_TheatersOfWar.pdf

3. Tim Lenoir, "All but War is Simulation: The Military-Entertainment Complex, *Configurations, 2000*, 8:289–335, The Johns Hopkins University Press and the Society of Literature and Science; available at http://humanities.uchicago.edu/sawyer/itsite/allbutwar.pdf

4. J. Calvin, A. Dickens, B. Gaines, P. Metzger, D. Miller, and D. Owen, "The SIMNET Virtual World Architecture," *Proceedings of the IEEE Virtual Reality Annual International Symposium*, 1993, pp. 450–455.

5. Bruce Sterling, "War is Virtual Hell," premier issue of *Wired* magazine, Mar/Apr 1993; available at `www.wired.com/wired/archive/1.01/virthell.html`

6. In addition to the cited references, inputs to this chapter came from Touraj Assefi (e-mail of March 13, 2003); Steve Blumenthal (e-mail of March 3, 2003); Jim Calvin (e-mails of March 13 and August 12, 2003, and e-mail of May 28, 2010 on which he consulted to Art Spear); Charlotte Hollister (e-mail of August 12, 2003); Drew Johnston (e-mails of August 29 and September 1, 2003); Dan Massey (e-mails of March 16, August, 29, and September 2, 2003); Duncan Miller (e-mails of August 13–15, 2003); Ray Nickerson (e-mail of March 14, 2003, quoting a memo of April 8, 1982); Dave Walden (memories while compiling the chapter); and JC Williams (phone conversation of August 2, 2003, and e-mail of September 1, 2003).

7. Dan Massey remembers once hearing Thorpe say that he got the idea for SIMNET from seeing four flight trainers that had been networked together.

8. Compiler's note: Duncan Miller is widely known informally as Duke Miller. Many people call him Duncan, and many others call him Duke. I will use Duncan except in quotations where people call him Duke.

9. Dan Massey, however, reports that throughout the history of the project, the people from Division 4 and Division 6 disagreed about the importance of the networking component and the use of standard networking protocols. In the end, SIMNET mostly brewed its own.

10. Delta Graphics was founded by Mike Cyrus and Drew Johnston with several of their colleagues from Boeing.

11. Dan Massey remembers how disappointed BBN people were with this decision. They initially had been given a target of $2,000 for the graphics subsystem. After many months of study, they believed we could produce something marginally acceptable for around $50,000. They were profoundly shocked when Thorpe chose the Bellevue group to create the image generator at a cost of $100,000. As it turned out, the $100,000 cost was optimistic for their design, which rolled out at a cost of $500,000. Once BBN people learned that Mike Cyrus — who led the Boeing group that conceived the image generator technology that they developed at Delta Graphics — was a longtime friend of Jack Thorpe, they stopped complaining about Thorpe's decision and got on the with part of the project they still had. Taking the long view, Thorpe's decision was undoubtedly the correct one for the SIMNET program.

12. There was considerable rivalry between the BBN and Perceptronics people, from the technical level up through the top management level. Not only did some BBN people resent losing control (and funding) of parts of the system to Perceptronics; they also didn't think highly of Perceptronics' capabilities to manufacture the training devices. BBN Systems and Technologies president Dave Walden made a point of personally visiting Perceptronics president Gersh Weltman each time Walden visited BBN's office in the Los Angeles region. Walden remembers these visits having been motivated by feedback from ARPA (perhaps from Thorpe himself) suggesting that BBN should get along with Perceptronics or face some undesirable consequence.

13. Although they regretted losing the graphics R&D, the BBN people generally had high regard for the capabilities of the Delta Graphics people.

14. Dan Massey says, "As to the origin of SIMNET, I guess that depends on what you think the critical innovation is. I have generally credited David Epstein with being the 'inventor' of SIMNET — that is, the person who saw how to hook the simulators together in a shared environment and first made it work. Of course, he was not alone, and his contribution was purely technical. Also, there was a high-level vision long before this."

15. Early on, some BBN people thought of Bloedorn as being an ally of Perceptronics (he used to ride his motorcycle from his home in El Paso, Texas, to the Perceptronics office near Los Angeles). Later, everyone on the BBN team came to admire Bloedorn, and all were deeply saddened by his tragic accidental death just as his SIMNET technology was changing the training world he loved. Bloedorn died in an accident near his home in El Paso. He was trying to help the driver of a truck that was having problems on a mountain road when the truck tipped and dumped its load of heavy pipes on him. As he had requested, Bloedorn was buried in the New Mexico graveyard memorializing the Buffalo Soldiers, the all-black 9th and 10th cavalry units that had fought in the Indian Wars in the 1870s and 1880s.

16. Miller also wrote, "Bloedorn had written a massive (many hundred page) document containing everything he thought we would ever need to know about armored warfare. I think we were handed several copies of this at our first meeting. He had been working on this for several months before our contact began on 1 April 1983. Perceptronics had already been under contract for a few months. I think Gary worked with Bob Jacobs and Jim McDonough to compile this massive book. The book became something of a joke among us, because Gary used to carry it around with him and would often drop it on a table at SIMNET meetings with a great thud, crying 'All you need to know is in here!'."

17. The following fall there was another demo at the AUSA (Association of the U.S. Army) Convention. The demo was in a hotel near Rock Creek Park in the Washington, D.C., area. This demonstration included the first, rough versions of image generators from Delta Graphics. At this demonstration Duncan Miller first met Jim Shiflett, who, Millers says "was the first Army officer I found who really grasped what was going on inside of SIMNET. At the first meeting, he presented an idea for how to incorporate mine fields into SIMNET that was essentially sound and workable. [Many other Army officers acted as if the simulators worked by black magic.] Shiflett was later pulled out of a battalion command in Europe by Gen. Thurman and sent to replace Thorpe as SIMNET program manager at ARPA. [He] subsequently became technical director of the Defense Modeling and Simulation Office and then the program manager for the million-dollar Close Combat Tactical Trainer (CCTT) program after contract award."

18. See Chapters 13 and 16.

19. See Chapter 6.

20. In an effort to also find synergy between the computer design groups in Cambridge (whose work is described in section 21.6, page 534) and Bellevue, Steve Powers spent a long time in Cambridge, Rich Johnston spent some time in Cambridge, and Randy Rettberg made several trips to Bellevue.

21. Jim Calvin remembers, "There was an anecdote at the first SIMNET demo that the total cost of development to that point in time was the same as the cost [would have been] had the same number of range rounds been fired as were fired at the demonstration."

22. The BBN team included (as remembered by Dan Massey and Dave Walden) Walden, division director Al Stevens, marketing person JC Williams, contracts person Ted Sihpol, Massey as prospective BBN program manager, and engineer James Chung.

23. JC Williams thinks having a separate team was useful to carry out the fixed-price AGPT contract; Duncan Miller thinks the original SIMNET developers were not allowed to participate or interact with the AGPT group and that this was an "astounding" mistake that resulted in the AGPT developers having to relearn lessons the SIMNET developers had already learned. Dan Massey says, "AGPT was programmed in Ada, a decision which caused a fairly acrimonious divide within the old SIMNET team, in which most of the original SIMNET developers (those who remained, like Calvin and Dan Van Hook, to say nothing of Duke) declined to participate (on account of severe Ada rejection and disagreement over the necessity for this change for AGPT). The AGPT simulators were designed to operate at the full frame rate of the BBN image generators, 15 frames per second. Because the contents of each frame were computed from 'Ground Truth' it was unnecessary to provide dead reckoning capability in the simulators. The requirement for 15 fps was an enormous speed and throughput increase from SIMNET. In addition, there were many more display channels in the AGPT Leopard 2 (main battle tank) simulator (14 compared to 6) each with about four times the pixel count. The Germans had seen SIMNET and thought it too unrealistic and 'toylike' for their purposes, which included use as a precision gunnery trainer. SIMNET was incapable of training gunnery due to the poor graphics, which represented a cost/performance compromise reached in the very early days of image generator design. AGPT led to the development of a greatly improved IG design by the Bellevue team, which was subsequently retrofitted into existing SIMNET sites."

24. Dan Massey says: "Basically, Marder went to Atlas Elektronik because of a political upheaval in the Bundeswehr procurement office that took Wegmann's influential man out of the picture and replaced him with someone dedicated to the other bidder. It is interesting that this team (Atlas Elektronik) had built the Bundeswehr's earlier sims and went on to build (as far as I know) most of their later ones, the AGPT Leopard 2 from Wegmann being the exception. Of course, Wegmann had a special interest in the Leopard 2, since they designed and manufactured the turret assembly for it, as well as a fancy trainer built on a motion platform of sorts. I believe that the 'other team' eventually bought image generators for their system from Lockheed-Martin, which acquired what remained (after Loral) of the BBN advanced simulation legacy.

"Later Wegmann was bought by another company that operates under the name of Krauss-Maffei Wegmann GmbH & Co. or just KMW. They are very active in building training systems, just nothing again much like AGPT."

25. Charlotte Hollister, who had spent years with BBN's PROPHET system (Chapter 12), moved to the Advanced Simulation Division in 1990 seeking new challenges. She first helped manage a project for the Army Research Institute for simulating various enhancements to the M1 tank; she then also became Dick Garvey's deputy, managing the original SIMNET work while many others in the division concentrated on trying to obtain new contracts. After Garvey died, she served as Greg Swick's deputy.

26. For the sale price, see Table 6.5, page 109.

27. Dan Massey believes that Loral probably could not have executed successfully the SIMNET ASDT contract had they not purchased BBN Advanced Simulation.

28. Compiler's note: I compiled the following account from a sequence of comments Dan Massey sent to me after he reviewed a draft of this chapter.

Chapter 21

Later Years of Basic Computer and Software Engineering

Compiled by David Walden

Chapter 4 covered some of the earliest computer operating-system and language work that was done at BBN. This chapter covers much additional language, operating-system, and computer design work that happened over the years.

Computer language and time sharing research and development, which had been part of BBN's earliest involvement with computers, remained important areas of work in later years (sections 21.1 and 21.2). Also, in later years BBN began designing its own computers and computer devices (sections 21.3 to 21.6), both for internal use and for use in larger computer systems sold to outside customers.

21.1 More high-level language work

BBN's early work with programming languages involved DECAL and the languages shown in Figure 4.1, one of which was LISP. This section sketches continuing LISP developments, development of the Parsec language, and work with languages for developing communications software.

LISP

The LISP programming language was invented by John McCarthy and has been widely used in AI work.[1,2,3]

The first LISP system at BBN was developed by Danny Bobrow and Dan Murphy. Bobrow had been part time at BBN since 1962, while doing graduate work at MIT. He did early important AI work in LISP at MIT. When he moved to BBN in 1965 to rebuild BBN's AI activities, decimated by the departure of Tom Marill (Chapter 16), Bobrow knew he needed a LISP system to do his work. He decided to build off of the in-core LISP for the PDP-1 that had been developed by 14-year-old high school student Peter Deutsch[4] and developed on a PDP-1 in the MIT lab run by his father, Professor Martin Deutsch.[3,5,6]

One of Bobrow's early hires at BBN was Dan Murphy. Murphy joined BBN in June 1965, immediately after getting his bachelor's degree from MIT. He knew Danny Bobrow from around the MIT AI lab. Murphy believes he began working on LISP on the PDP-1 right away (and in later years he worked on LISP for the SDS 940 and PDP-10). Bobrow and Murphy worked to develop an extended memory LISP (using the drum memory on the PDP-1), to provide a large enough address space for the AI work. From the time Danny Bobrow joined BBN full time, having a good LISP programming infrastructure was important to BBN's AI people and some others as well. On some occasions, creating a good LISP environment drove other computer developments. Bobrow and Murphy started with the LISP 1.5 definition and looked at Deutsch's code for the PDP-1b, but their implementation had a completely new code base for what they called BBN-LISP.[7] Murphy says,

> . . . one of the things we did was build a timesharing LISP system on the PDP-1 — that is, a multi-user system for which the primary (and only) interface language was LISP. It was both the command language and the programming language, a la JOSS and such systems of that day.

In their PDP-1b implementation of LISP, Bobrow and Murphy grappled with issues of supporting a large LISP memory on a machine with a small physical memory.[8] In particular, Murphy implemented a software-based virtual memory and demand paging system[9] within his LISP implementation, and Bobrow and Murphy studied its performance.[10]

Later Danny Bobrow encouraged acquisition of an SDS 940 time-sharing system (see section 21.2) to provide a better LISP environment.[11]

Another relatively early hire by Bobrow was Warren Teitelman, a PhD student of Marvin Minsky at MIT (like Bobrow himself) and a onetime roommate of Bobrow. At BBN, Teitelman was renowned for his lightning fast programming, including the speed of his typing into the computer. Much of his work at BBN was aimed at extending LISP[12] and improving the LISP programming environment to enhance productivity generally; for example, he added a spelling corrector and the Do What I Mean (DWIM) package that automatically discovered and corrected typical programming errors.[13,14]

According to Steele and Gabriel,[3]

> [Teitelman's BBN-LISP] . . . introduced many radical ideas into LISP programming style and methodology. . . . The origin of these ideas can be found in [his] PhD dissertation on man-machine symbiosis.[15] In particular, it contains multiple extensions to the notions of Deutsch's on-line structure editing (as opposed to "text" or "tape" editing[16]), breakpointing, advice . . .

When there was no good successor time-sharing system to the SDS 940, Bobrow pushed for BBN to develop its own system, TENEX (see section 21.2, page 523), and the BBN-LISP system was ported from the 940 and extended on TENEX.[17]

In the years since LISP 1.5, many variations of LISP have been developed, e.g., MACLISP, SCHEME, Franz LISP. Each of these LISP systems had more or less subtle differences from the others: All the issues that distinguish any programming language distinguished LISP variants (e.g., issues of data types, function call argument evaluation); and there were distinctions that were more unique to LISP at the time (e.g., methods of providing a virtual memory and garbage collection, degree of integration of the development environment with the programming language).[18]

BBN-LISP was one of the more well-known LISP variations, undoubtedly because of Teitelman's extensive development efforts[19] and because BBN-LISP came with TENEX and its virtual memory support for LISP. Also, unlike many others, it was a production-quality LISP system. At one point, BBN was supportive of BBN-LISP's being ported to the IBM 360 class of machines at the University of Uppsala, in Sweden; Alice Hartley provided expertise about the LISP system at BBN to the people at Uppsala doing the port.[20]

In time, Bobrow and Teitelman joined a stream of BBN people moving to Xerox PARC. At PARC Teitelman continued his LISP development work, BBN-LISP was subsumed under the name InterLISP. BBN and PARC jointly maintained InterLISP for a while, with PARC primarily handling the programming environment and extensions to the language and BBN primarily maintaining the kernel.[21,22,23] Eventually PARC took over ongoing maintenance and development of InterLISP,[24,25] BBN's involvement in developing InterLISP began to be forgotten by people who joined the LISP community after the early

1970s, and BBN mostly moved to the sidelines of the active LISP development world of the 1980s and beyond.[26,27]

As the workstation era drew near, BBN developed its own workstation, called Jericho (see section 21.4, page 528), and InterLISP was ported to this machine (other groups also ported it to other systems, e.g., Xerox PARC to the Alto[28]). In turn, BBN made extensive use of Symbolics machines for LISP (at first running the ZetaLISP version of MAC LISP, later Symbolics Common LISP), and then Franz LISP running on Sun workstations. Some Mac-based projects used Macintosh Common LISP.[29]

PROPHET and Parsec

In 1968, as the Hospital Project (see Chapter 4) was getting smaller, people in Frank Heart's division sought new contracts. Paul Castleman was having discussions with Bill Raub of the National Institutes of Health about BBN's building a system to help medicinal chemists and research pharmacologists use time-shared computers to help them with their research and development. Ultimately, the system BBN built was called PROPHET and was an enormously successful contract for BBN for many years. Paul Castleman describes this in his companion chapter (Chapter 12).

At some point Paul took me along to visit Bill Raub. Thus, in response to the desire to offer Raub something unique that BBN could build, I wrote a position paper[30] explaining why the system we were discussing should be based on an extensible programming language (extensible languages had recently become a state-of-the-art idea[31]) so that chemists and pharmacologists could have chemistry-related data types and operations in the programming languages in addition to the typical algebralike constructs and data types in languages like FORTRAN. To show the feasibility of this, I obtained the FORTRAN listing of James Bell's PhD thesis on the Proteus extensible language and spent a few days transliterating it line by line into PDP-10 assembly language.

We probably claimed some sort of milestone with this implementation, and it may have helped us with Raub in some way; however, the implementation never really worked. Fred Webb was asked to take over maintenance of my implementation. He doesn't know how much, if any, of what I did got preserved. The implementation philosophy changed completely at an early point in his involvement, and at that time he may well have started again from scratch.[32]

In any case, Webb's version of Proteus was named Parsec and Parsec was instrumental to the success of the PROPHET system. Of Parsec, Fred Webb said,[33]

> Parsec was derived from Proteus, the subject of a PhD thesis from Jim Bell, who joined Digital Equipment Corporation after he received his PhD. Proteus was an extensible language, allowing complete user-defined syntax.
>
> The unique thing about the design of Proteus was that the syntax description language was changed into a syntax programming language, complete with standard programming language constructs.
>
> The implementation of Proteus was done entirely in PDP-10 assembly language. The code which drove parsing was very tightly coded. I believe that I did the initial implementation, which was then optimized by myself and Steve Butterfield. The result was something that allowed a user-defined syntax description to be fast enough to be practical. The syntax description language was not compiled into machine code, but was interpreted from some optimized data structures that were created from the parser description.
>
> The only use of Parsec was the implementation of the PROPHET system, which included application specific data types = tables, molecules, graphs and the PL/PROPHET programming language.

Other Language Efforts

As the years went by, BBN increasingly used language processors developed by vendors or other institutions rather than developing them itself. BCPL, C, C++, and Java have all been extensively used. Still, there there were some new language development efforts.

Bob Morgan and Art Evans were involved in a sequence of language development efforts.[34] Bob Morgan says,[35]

> The idea began with the Pluribus IMP work which was difficult to program. The question was, "How to get a high-level language to program communications software." At that time the Defense Communications Agency was looking for someone to study it, and Art Evans and I wrote a proposal which was funded.[36] Art and I designed the language and compiler. Art led the language design and I led the compiler design. The language was called COL (Communications Oriented Language). It was the basis of our proposal to be one of the four Ada language design teams — we lost. It was also the basis for the PRAXIS programming language we designed for Lawrence Livermore. There we did the modification of the language design and implemented the compiler/runtime on the PDP-11 and the VAX. The purpose of PRAXIS was to program the LSI-11s and the VAXs used in the Nova-Shiva Laser Fusion Project.
>
> At the same time we were consultants to the Defense Communication Agency to advise them on DOD-I, which is Ada. Art was on one of the major committees.

There undoubtedly have been other language development efforts over the years at BBN that we won't describe in this section.

21.2 More time sharing

BBN's time-sharing system development work (the early work was described in Chapter 4) continued to be innovative as time-sharing systems became more powerful.[37]

SDS 940

The SDS 940 time-sharing system was developed at U.C. Berkeley with sponsorship from ARPA.[38] According to Butler Lampson's website, about 60 SDS 940 machines were sold; it was marketed by SDS (later XDS).

In the late 1960s, Danny Bobrow was pushing LISP, working closely with Dan Murphy. His division got an SDS 940 partly so Bobrow and his people could have a better LISP environment than Murphy had been able to create on the PDP-1b.[39] The SDS 940 became the major computer resource for much of the company (although many of us in Frank Heart's Computer Systems Division continued to do some of our production work on the PDP-1d). Ted Strollo managed the Research Computer Center where the SDS 940 replaced the PDP-1b.[40]

Instead of the mag-tape-based file system that was originally part of the SDS 940, the installation at BBN had a giant (literally) hard drive unit[41] from Bryant Computer Projects (a subsidiary of the Ex-Cello Corporation, a food-packaging machinery company). It had 26 disks, each of which was one meter in diameter. It was subject to head crashes that would destroy multiple disks. Ray Tomlinson remembers,[42]

> The source of the head crashes was dust or magnetic oxide particles that would pass through the microscopic space between the heads and the disk surface and scrape the surface producing more particles. This would cascade until eventually all the disks on one side (13 platters) would be wiped out. A particle detector retrofit attempted to minimize the cascade effect by sensing the particles and forcing an emergency head retraction. This often reduced the number of platters affected to only one or two.

I remember one of these shiny one-meter metal disks being used by Jerry Elkind as a coffee table for his office.

Perhaps because of the unreliability of the 940 disk drive but also because many users were leaving their programs and data on the disk drive, Dan Murphy and Ray Tomlinson created a sophisticated program to maintain copies of all disk files on magnetic tape.[43] The people in Bobrow's group also put up a major LISP system on the machine.

In addition to providing a useful computing resource for BBN, the SDS 940 system also provided useful ideas to BBN's TENEX development (see below). Ted Strollo reports that the BBN people were in close contact with Butler Lampson and the SDS 940 software people at Berkeley, since they provided the software support (you got the hardware from SDS and the software from Berkeley) and they, like BBN, were part of the community of ARPA research contractors.[44]

TENEX

In time the SDS 940 system became inadequate for BBN's AI work. In particular, the memory limitations of the 940 time-sharing system got in the way of running big LISP programs.

When there seemed no good successor to the 940, Bobrow pushed the idea that BBN should develop its own operating system. With both Bobrow and Jerry Elkind promoting it, ARPA funding was obtained, and the TENEX system was developed. Ted Strollo managed the project and participated in various aspects of the design and implementation. Jerry Burchfiel,[45] Dan Murphy, and Ray Tomlinson were senior members of the development team. Bobrow himself participated in the system specification. Others who participated in the project in some way over time included Don Allen, John Barnaby,[46] Ed Fiala,[47] Bob Clements, Elsie Leavitt, Tony Michel, Ted Myer, Bill Plummer,[48] and Don Wallace.

Tomlinson recalls the decision to develop TENEX:

> . . . there were not many choices. We had experience with the Scientific Data Systems 940 that we were using, and it was inadequate for the natural language researchers (LISP). Multics sacrificed a lot by being written in PL1, had a klunky user interface, and was trying to be more than we needed. The PDP-11 virtual memory space was not big enough to run big LISP applications, if in fact virtual memory versions of the PDP-11 were available at that time. The SDS Sigma 7 was also inadequate. The DEC PDP-10 had a good track record in running LISP (natural 18-bit addresses), but required huge amounts of real memory to run well because its memory management support was negligible. So we decided to build our own virtual memory hardware and write our own operating system.

Bobrow also notes,

> Commercial vendors of the time didn't believe in large virtual memories, so what BBN needed was unavailable commercially. Also, Multics was both expensive and long in the future.[49] Thus, a reasonably priced approach was for BBN to develop a pager box for the existing PDP-10 with its large address field instruction.

As the project began, Bobrow remembers, the BBN team spent a lot of time thinking about the problems that they had encountered in using other operating systems — and about ways to get around them. For instance,

- file versions so that one didn't delete files by accident, could maintain multiple versions, and could have automatic deletion strategies

- executive commands that were mnemonic, with command completion as a way of understanding what you were doing and typing minimally
- user control, such as use of control-C to interrupt a program at any time
- easy separation of the operating system kernel from applications
- ways of having applications communicate
- multiprocessing for individuals and a good scheduler that balanced overhead and response

The TENEX paper[50] divides the goals into three categories: state-of-the-art virtual machine;[51] good human engineering throughout; and an implementable, maintainable, and modifiable system.

The TENEX design and development were very successful, and the details of the TENEX design have been well documented.[50,52,53] Its deployment also was successful. In addition to being an excellent time-sharing system, TENEX also ran all of the DEC TOPS-10 software (editors, assemblers, compilers, debuggers, etc.), implemented via a compatibility module that emulated the necessary parts of TOPS-10 under TENEX.

BBN exploited TENEX is several ways. First, the Research Computer Center provided TENEX service to internal users. Second, BBN sold time on TENEX externally, over the ARPANET, to users who didn't have their own computer facilities. Third, BBN provided TENEX systems to other sites (in somewhat the same way as Berkeley had provided the SDS 940 to BBN). Ted Strollo managed all these activities.[54]

Because of the electronics technology of the time, the pager was housed in a two-foot-wide, six-foot-high rack. It was an interface between the PDP-10 processor and the memory bus and provided a quite sophisticated demand paging function to support large virtual memory spaces for processes including controlled sharing of pages. The pager boxes were built by and at BBN and wired to PDP-10s at customer sites by BBN technicians.

The combination of the PDP-10, the BBN pager, and the TENEX software provided the first practical virtual-memory computer time-sharing system. On the order of a dozen ARPA research contractors ordered a PDP-10, bought the BBN pager, and had BBN install TENEX. Thus, TENEX systems were quite prevalent among the early computers on the ARPANET (BBN also provided the TENEX-ARPANET interface in many instances), making TENEX important in early Internet history. In addition to much useful research and development that was done on TENEX systems, at least one great moment in computing happened on TENEX: BBN had two TENEXs in house (one for BBN users and one for TENEX development); and Ray Tomlinson took this opportunity to hack together the first instance of networked e-mail (see Chapter 18).

In my view, and I am far from alone, TENEX was a great improvement over its predecessors[55] and was better than any operating system I have used since (e.g., UNIX, DOS, Mac, MS Windows). UNIX inventor Ken Thompson said, when he received the ACM's Turing Award,[56]

> I can't help but feel that I am receiving this honor for timing and serendipity as much as technical merit. UNIX swept into popularity with an industry-wide change from central mainframes to autonomous minis. I suspect that Daniel Bobrow[52] would be here instead of me if he could not afford a PDP-10 and had to "settle" for a PDP-11.

In 1973 BBN sold the rights to TENEX to DEC, and Dan Murphy switched his employment to DEC to help DEC convert TENEX into what became TOPS-20.[9] TOPS-20 was released by DEC in 1975–76 and had a long successful life.[57]

Figure 21.1. TENEX pager and DEC-KA10 machines used for TENEX development and time sharing, circa 1970. (Photos courtesy of Dan Murphy.)

Accounting System Research

In addition to its technical innovations, BBN implemented a time-sharing system accounting innovation known as the pie-slice scheduler.

At the time, circa 1974, time-sharing service typically was sold on the basis of some combination of connect time and resources used (e.g., machine cycles, pages of disk space). However, if you are doing computer research (or any of numerous other kinds of research), minimizing computer use to minimize computer charges (and thus avoid project cost overruns[58]) is counterproductive to the research you are trying to do.

As manager of the computer center, Ted Strollo felt the anguish of users faced with the detrimental cost and technical results of traditional use-based accounting. An idea emerged to level the amount of money spent each accounting period (half month) by selling the machine on a percentage or slice basis. This idea was implemented in the so-called pie-slice scheduler. At any time, the scheduler allocated machine resources among users from various projects and departments in proportion to the shares of the machine each project or department had bought, averaged over perhaps two-week intervals). Don Allen crafted many of the pie-slice scheduler refinements to Dan Murphy's original TENEX scheduler.[59]

21.3 Adapting existing computers to packet switching

In 1969, BBN's development of the ARPANET packet switches involved additions to existing computers that made them more suitable for real-time operational tasks — which, in turn, got the engineers and programmers thinking about computer design more generally.

516 IMP, TIP MLC

In 1969 BBN developed the 516 IMP packet switch for the ARPANET. This was a combination of a Honeywell 516 minicomputer; special interfaces (with general design by Severo Ornstein, detailed design and manufacturing by Honeywell, and cleanup of the Honeywell work by Ornstein and Ben Barker[60]); and software developed by Will Crowther, Bernie Cosell, and me. The story of this development has been widely told[61,62,63] and is touched upon in Chapter 17. In 1971 BBN developed the ARPANET TIP, which was based on a Honeywell 516 or 316 computer with a BBN-designed (by Ornstein with help from Barker and Tony Michel) and -manufactured multi-line-controller (MLC) with software (developed by Crowther) to allow up to 63 terminals to connect directly to an ARPANET packet switch.[64] This story, too, has been widely told[61,63,65] and is mentioned in Chapter 17. The IMP and TIP hardware had to interface between the analog world outside and the digital world inside at the real-time rates of the external world. The software included innovative work on dynamic reconfiguration of network lines and nodes.

Lockheed SUE and Pluribus

The dynamic reconfiguration developments of the ARPANET got the BBN engineers thinking about how to make a packet switch that itself was fault tolerant. This led to development of the Pluribus computer based on a combination of Lockheed SUE minicomputer components and BBN designed (by Severo Ornstein, Mike Kraley, Tony Michel, and Ben Barker) and -manufactured components with lots of innovative scheduling and reliability software (initially designed by Will Crowther). This project is described in more detail in section 21.6. Randy Rettberg joined the team to work on the Pluribus-based Satellite IMP.

PLI and B-C-R

Since common-user, packet-switching computer networks like the ARPANET (and Internet) have data from many users flowing through the same switches and circuits, security issues arose that had not been present in earlier end-to-end security systems using dedicated circuits. This led to the development of the Private Line Interface (PLI) device, the B-C-R (for Black-Crypto-Red) device, and the IPLI (Internet PLI) device. These devices provided the first illustrations of how security might be provided in a modern networked environment.

21.4 Building complete computer systems

In the late 1970s and early 1980s, because microprocessor chip sets had become available, it was feasible for BBN engineers to build computer systems that the company could call its own rather than building systems based on existing computers as was done with TENEX (section 21.2) and the packet switches (section 21.3). Three “BBN systems” were built: the MBB and Jericho computers and the BitGraph terminal. There were technical similarities among these systems, but each had a different motivation (other than the motivation of engineers wanting to do state-of-the-art work), and they were quite separate projects organizationally (an aspect of “the BBN way”).

MBB

The Microprogrammable Building Block (MBB)[66] architecture[67] was the idea (hatched in 1978) of Mike Kraley and Randy Rettberg, both of whom had previously been doing hardware design for the Honeywell 316-based machines or Pluribus machines that were involved in various aspects of the ARPANET and in the beginning experiments with interneting (there is more about this in Chapter 17).

Microprogramming was far from a new concept; however, a key and more unusual goal of the MBB was to provide for "user" microprogramming so that different user groups could microprogram the machine to suit their own needs. This user microprogrammability was provided through an unusually simple microcode machine design, including a simple 32-bit instruction format that did one thing at a time; availability of microprogramming support tools;[68] and a large, dynamically alterable control store.[69]

Randy Rettberg explained that he and Mike Kraley conceived the MBB for several reasons. First, Honeywell was going to stop building the Honeywell 316 machines on which BBN had been running its packet-switching software (see also Chapter 17). The Honeywell 716 didn't look like a good option for the packet-switching software: BBN's Telenet start-up had originally ported the IMP code to the Honeywell 716 and had not been happy with the I/O system. Rettberg had looked at the possibility of automatically porting the IMP code to another machine, and it looked hard. Therefore, a new machine had to be found. Microprogramming a microcomputer to emulate the Honeywell 316 offered the same performance of the Honeywell 316 at half the price and did not require an immediate rewrite of the packet-switching software. Second, a government customer wanted a smaller, less expensive packet switch. Finally, while sitting around their hotel rooms at an International Solid State Circuits Conference in San Francisco in February 1978, Kraley and Rettberg noted that (at least their part of) BBN was not then doing any new hardware projects. So they decided to get their division (Frank Heart's division) to initiate projects to design and build both the MBB and the Butterfly (section 21.6) computers.

Thus, the MBB project was undertaken, using a standard 74S181 ALC that allowed use of 1,024 register windows such that they could be quickly switched in response to a low-level interrupt. The first goal was to emulate the Honeywell 316 computer. After the first design review for the implementation of the 316 instruction set on the MBB, the team saw that they could implement a 316 with 20-bit rather than 16-bit words (which the Honeywell 316 used), enabling a larger address space that would extend significantly the life of the IMP software written originally for the Honeywell machines. Consequently, they chose 20 bits as the memory word width for the MBB, a size usefully bigger than 16 bits but not so big (e.g., 32 bits) as to double the amount of hardware needed. All data paths, registers, and the arithmetic logic unit were constructed to be 20 bits wide. When emulating a 16-bit machine, the extra 4 bits optionally could be ignored, or they could be used as they were in the MBB-based packet-switching software.

The MBB also had interesting designs relating to a register file, flexible emulation, memory subsystems, microcode organization, and input/output and interrupts. For instance, the MBB had a connector between the instruction register and the microcode instruction dispatch to implement a perfect hash lookup. Another connector was placed between the memory address register and the memory address lines. These connectors allowed configuration of the machine for different instructions sets and applications.

While Rettberg and Kraley jointly conceived the MBB and Butterfly projects, eventually a meeting was held with division director Frank Heart to divide and focus the work. Kraley took responsibility for the MBB, and Rettberg took responsibility for the Butterfly.

By the fall of 1979, the MBB was operating as a packet switch. This conversion of the Honeywell 316 packet-switching software to the MBB was very successful;[70] the new packet switch was called the C/30, and many C/30-based networks with a variety of characteristics (X.25 compatible, TEMPEST-tested, etc.) were installed.

In 1979 BBN created a product business known originally as BBN Computer Corporation (BBNCC), with Ben Barker as its president.[71] At the outset this business did the ARPANET hardware maintenance, which Barker had been managing in Frank Heart's division before BBNCC was created, and handled construction of the Pluribus systems (section 21.6). BBNCC also had ambitions to sell a UNIX system on BBN hardware. Thus, a version of the MBB microcode was developed to create a computer that could more or less directly execute the C language in which UNIX was written. It was called the C/70. Some C/70-based UNIX systems were sold, but in the long term that business did not succeed. The MBB was built at the end of the 16-bit DEC 11/70 era, leading BBN to think it could compete with DEC. Unfortunately, DEC was about to release the VAX 11/780 and other 32-bit machines.

Eventually BBNCC was renamed BBN Communications Corporation and utilized the C/70 as one component in network applications. The total number of MBB-based systems deployed approached perhaps 1,500 (C/30s and C/70s).

Jericho (BBN Doesn't Start a Workstation Company)

To some extent, the Jericho workstation[72] is another example of BBN people's not being able to wait to have what they needed. Ray Tomlinson says,

> Jericho was developed on IR&D[73] because personal computer workstations were not yet commercially available. Our researchers needed personal computers that could really compute, not the toys on the market by that name.

Harry Forsdick remembers that Jericho also was developed partly to show ARPA that BBN remained a leading place for research in distributed systems (this was a time when Xerox PARC was getting lots of attention and quite a number of researchers had left BBN and gone to PARC).

Tomlinson continues,

> [Jericho] used a bit-slice architecture and ran microcoded interpreters for either Pascal or LISP. It had bit-mapped monochrome or color graphics display, keyboard, mouse, hard drive, fiber-optic network interface,[74] sound chip, and ASCII terminal interface. It was housed in a 30 inch high rack.

Jim Calvin remembers,

> . . . the first full Jericho was running in early 1980 (or perhaps late 1979). Ray [Tomlinson] and I did the bulk of the hardware [Tomlinson did the main processor board and bitmap display board, and Calvin did the main memory and IO board]. Alice Hartley [and Norton Greenfeld and others] did the InterLISP system. Harry [Forsdick], Ray, and others [e.g., Jim Miller] also worked on the Pascal implementation. [Harlan Smith and Art Spear worked on mechanical issues.]
>
> The Jericho systems were designed, at least in part, to support research into distributed systems. . . . [O]ther than being large by today's standards, you would still recognize a Jericho as a workstation . . . — windowing system complete with pop-up menus, etc., virtual memory — it was all there. I remember sometime after the Mac II came out, perhaps 1988/9, thinking I now have a machine and software environment that finally is as good as the Jericho I was using in 1982/3.

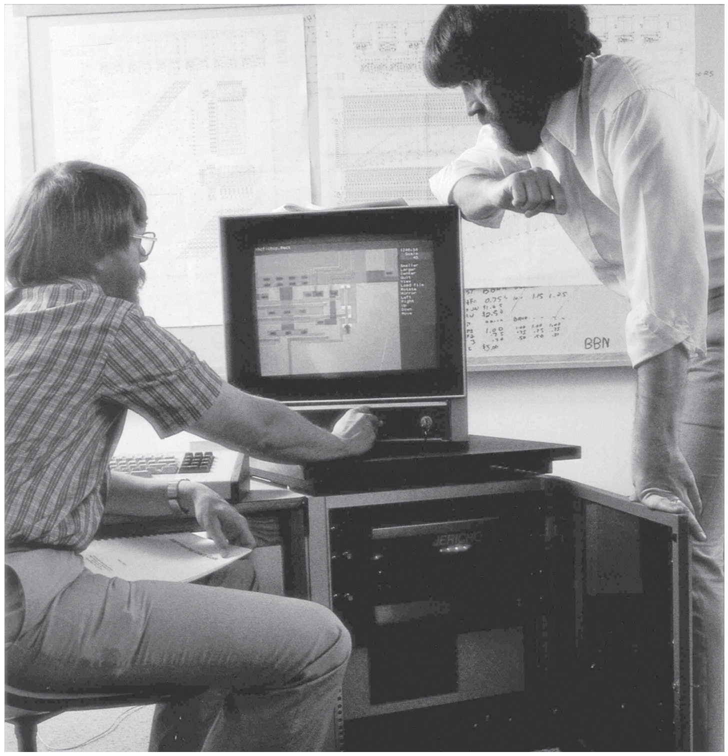


Figure 21.2. Ray Tomlinson and Jim Calvin working at a Jericho workstation. (Photo courtesy of Jim Calvin.)

> The sound chip Ray mentions was actually a 6802 based system that could generate specific sounds much like arcade games of that era (this worked well for the version of PacMan that ran on Jericho). It also had a pass-through mode that allowed us to pass 8-bit digital samples through to play back recorded voice or anything else we desired. Jericho had 3 serial interfaces (Signetics 2651s): one for the keyboard/mouse, one for a terminal, and one for whatever. . . . the hard drive was approximately 200MB which was extremely large for 1980.

Harry Forsdick expands on Tomlinson's and Calvin's memories:

> The Jericho hardware was built from the AMD 2903 bit-sliced microprocessor, a microprogrammed device on top of which we developed two virtual machines: InterLISP and Pascal. The InterLISP VM supported the variant of LISP favored by BBN and Xerox PARC and was primarily used by Artificial Intelligence research groups at BBN building natural language understanding systems.[75] The Pascal VM[76] supported the UCSD variant of Pascal, and was used by groups at BBN developing early single-user workstation environments. (Remember, this was before PCs and Macs). Innovative ideas in the Jericho software included a bitmapped, windowed programming development environment with integral debugger, an implementation of Cronus, a distributed resource system, and the Diamond multimedia communication system, the first integrated multimedia email and workspace conferencing system.[77]

Division director Ray Nickerson and colleagues talked to BBN president Steve Levy about going into the workstation business; but the business case was not strong enough at the time, and Jericho remained an internal BBN project. About 28 Jerichos were built and deployed within BBN.

Nonetheless, the original goals for Jericho were more than met. As Jericho became available, BBN was awarded millions of dollars in research contracts that used Jericho as their computing platform. Jericho was also a useful credential for, and platform that was used in, several additional distributed systems contracts (see Chapter 18); and it brought attention to BBN as a place where additional government funders could sponsor leading-the-art research and development in, for example, distributed logistics systems. Forsdick says,

> In essence, Jericho catapulted [BBN] from thinking about tightly linked applications running on time-shared computers to understanding the implications of many autonomous, distributed personal computers working together on a problem.

BitGraph (BBN Doesn't Start a Workstation Company, Again)

Mike Kraley initiated the BitGraph IR&D project and oversaw it.[78] The design began in January 1981. As Robert Wells remembers, Phil Carvey did the hardware design,[79] and Wells, Dave Taenzer, and Dave Barach wrote the initial software.

Wells says that the software was

> written in very tight 68000 assembler — we wanted to make sure it would be fast enough. ... I remember being personally responsible for the "Rastop" (i.e., "BitBlit") code that would copy and transform rectangular regions between screen and/or memory, including font memory. . . . I used and abused every conceivable register in the 68000 to minimize memory references and speed up the inner loops — it was the high water mark of my assembly programming days.

Among them, Barach, Taenzer, and Wells also wrote code to make the BitGraph support VT100 and ANSI terminal compatibility, Tektronics 4010 emulation, the downloading of variable-width fonts, the scrolling of rectangular regions, and mouse input. At the same time as the BitGraph development, they were developing the PEN text editor, so they made sure PEN could make good use of the arbitrary-rectangular-scrolling capability for scrolling multiple text windows up and down and left and right.

During BitGraph development, Barach and Taenzer also created an extended version of the PEN editor that had a multipaned source code debugger that communicated over the RS-232 line with a remote debugging kernel. Wells recalls that he and the others

> used this to debug the BitGraph code, running PEN on another system such as a VAX, and using a terminal line to connect to the development BitGraph, and being able to set breakpoints visually, and have variable values automatically updated. It was a very visual debugger environment in early 1981 before it had been done much (or at all?) elsewhere. This environment was later used for remote debugging of some other BBN projects.

In the late spring of 1981, Barach and Taenzer left BBN to help start one of the first UNIX business application companies with Ned Irons, and Wells moved to other BBN projects. Wells says, "I continued to use a BitGraph in the office for several years and then at home for several years more."

Rich Fortier and his team at BBNCC took on responsibility for BitGraph software. They rewrote a lot of it in C for easier maintenance. Wells believes they added fancier sound support and the capability to download code to the BitGraph. BBNCC also took over the manufacturing of BitGraph.[80]

Because I was in management by that time, I never had a BitGraph — they seemed to be the province of the engineers and programmers. What I remember most about BitGraph is people playing Pac Man on it. Wells says,

> PacMan was a gorgeous hack — I'm pretty sure there was a version of missile command as well, where you frantically moved the mouse around trying to blow up the missiles in mid air.[81] ... There were also a lot of neat sound hacks — it had a good sound chip in it, and I think it was exposed both through escape codes and through downloaded code.[82]

By the time BitGraph was solidly in and being sold by BBNCC, Chuck Stein had moved there from the BBN corporate business development position and was leading the marketing of BitGraph. No one I have talked to is sure how many BitGraph systems were built and sold in total. At one point 30 to 40 a month were being built, so perhaps many hundreds or even a thousand or so were built and shipped in total.

BitGraph was fairly well known in the world at the time, and there was even a Usenet discussion group for it.[83] As of 2003, BitGraph terminal support was still included in most UNIX and Linux distributions (see, for example, the `/etc/termcap` file in Red Hat Linux), and BitGraph mouse support from John Robinson existed in Emacs distributions (`lisp/term/bg-mouse.el`).

Dave Mankins and Dan Franklin developed a sophisticated windows manager for the BitGraph;[84] namely, the capability of multiple windows stacked on top of (or beside) each other with the ability to trivially switch which the visible window or windows. The BitGraph Window Manager used the BitGraph's dynamic font-definition capability to store a window and its contents as a single character. To make the window and contents visible, that character was transmitted to the display over the RS-232 BitGraph input. The window manager supported different processes on the UNIX host computer having different windows on the BitGraph.

BitGraph stayed a terminal, although it had potential to be a full-scale computer. Rettberg thinks that

> one of the reasons we didn't put a disk in the bitgraph was that, if we did, it would be a computer. It was ok with Frank Heart if it was just a terminal, but not if it was a "computer."

Nonetheless, in those early days of workstations, there was much musing about doing more. Wells says,

> I have a vivid memory of all of us standing around a bench one day, looking at spec sheets and drawings for the Sun-1, and talking about how easy it would be to turn the BitGraph into a UNIX workstation — just give it a memory management chip, an Ethernet port, and an optional disk, and do another early BSD UNIX port to it.

Mike Kraley remembers traveling to the West Coast with Chuck Stein to see who they could interest in BitGraph.

> We met with Bill Joy, who was polite, but not terribly interested, but suggested we meet with Andreas Bechtolsheim, who was a colleague of his and more of a hardware guy. We met with him the next day, in a tiny attic office under the eaves somewhere in Berkeley. He was quite interested, and asked lots of questions about what we had done and compared notes. It seems he had been working on a very similar project, but was nowhere near as far along as we were. His design was actually very close to ours in many respects. He and his colleagues had been trying to convince the university to sponsor them, but that wasn't going well, and they were mulling whether they should strike out on their own or try to make a deal with an existing company. He thought the idea of partnering with BBN might be very exciting, since we were a 'real' company, had a manufacturing facility, support, etc. We left, elated, with promises to continue our discussions. About two weeks later, we learned that Bechtolsheim and his colleagues had indeed formed their small startup — Sun.

Figure 21.3 BitGraph terminal. (Photo courtesy of BBN.)

Robert Wells remembers that eventually BBN licensed the BitGraph design to a company that made a version of the display that rotated between vertical portrait and horizontal landscape modes, and later BitGraph turned up in Japan with Japanese fonts. Bob Brown[85] recalls this era in BitGraph's history:

> BBN sold the design to Forward Technology Inc. (FTI) who made a clone of it. FTI was bought by a Japanese company called DCL who repackaged the whole thing into two boxes — a separate processor box and a funky terminal. DCL was then bought by another company who jettisoned FTI. DCL/FTI had a liquidation sale and had an inventory of about 100 BitGraph clones that they were selling off for $100 and $200 each, depending on the memory size. I bought 40 of the $100 versions. . . . I rented a truck and picked them up and took them home. The shipping boxes (a 30 inch cube and an 18x18x30 inch box for each of the 40 BitGraph) completely filled a large room in my apartment — floor to ceiling. I then sold them off one by one for $500 (decreasing to $200 over time) to whoever would buy them. A fair number went to the EE department at Purdue University. A friend at NASA bought a few, too. . . . Profits from that — not much by today's standards — helped me make the down payment on my first house.

Robert Wells concludes,

> BitGraph was certainly one of my favorite projects at BBN, particularly in terms of the fun-to-time-spent-on-it ratio — we were only on it for a few months, but had a lot of fun with it, and did some really neat stuff. BitGraph was probably the best terminal ever.

The activity of researching this section demonstrated the fondness programmers and users felt for BitGraph. As they were found and began to exchange memories, some of them went into the attics or garages and found their old programs and manuals and excitedly discussed them with each other. Bob Brown found and fired up an actual BitGraph (or its FTI copy), ran programs on it, and posted photos to the web, while the others cheered him on by e-mail. Brown's final remark: "Seeing the [BitGraph] boxes literally sent chills down my spine. . . . Let's do this again in 20 years."

21.5 Computer devices

In addition to working on improving the operating systems and languages of more or less existing computers, over the years BBN was involved in a lot of devices that attached to computers. A few examples will suffice.

Data Equipment

As mentioned in Chapter 6, BBN had a Data Equipment Division located in Santa Ana, California. According to Dave Keast[86] who managed the division,

> BBN Data Equipment didn't have too much to do with computing. It was originally set up to build analog (Plotamatic) X-Y recorders and this remained its main business. We also built several Rand Tablet (Graphicon) devices for computers (which cost $10,000 then). We also built an analog rho/theta graphic input device and an opaque projector designed in Cambridge that projected Model 33 Teletype output onto a screen using some fancy image-reversing optics.[87]

Fibernet

In about 1980, the same BBN people who were developing the Jericho workstation (described in section 21.4), developed a local area network. As happened so often, BBN people couldn't wait for commercial developments. John Robinson remembers[88]

> It was called Fibernet. It used fiber-optic connections running at a modest speed, like 2 mbps. It evolved at the same time as the Jerichos. It was rapidly overtaken by Ethernet, but had its origins at about the same time. I don't think BBN had any illusions about a product here, just the desire to get something fast working in house.

Bob Clements remembers the details of Fibernet[89]

> Fibernet started out under the name "CheapNet." It was BBN's first local area network, and came into existence in the context of Chaosnet at MIT and Ethernet from DEC/Intel/Xerox. Both of those required (at that time) quite a complex and expensive controller and network (analog) interface to talk to the shared coaxial cable.
>
> BBN decided to try to make a cheaper local network. [Jerry] Burchfiel, [Bob] Clements, [Ray] Tomlinson, et al., came up with a simpler design, with one important "sexy" feature.
>
> The "Cheap" feature consisted of using the fastest off-the-shelf serial interface we could find to send and receive packets. This turned out to be the Signetics 2652-1 "speed selected" HDLC USART chip to carry HDLC packets with their inherent addressing bytes and CRC error detection. This ran at 2 Mbps. IP packets were minimally encapsulated in HDLC frames.
>
> The CheapNet fiber infrastructure was installed across all of the multi-building BBN Cambridge campus. It supported dumb terminals, Jericho workstations, and central timesharing machines.
>
> There were four hardware components to the system:
>
> 1. a fiber transceiver board (called a tap) with interfaces (via copper) to the other components and via fiber optics to other hubs
> 2. the internal network interface of the Jericho workstation
> 3. a single-board terminal concentrator supporting 16 RS-232 terminals[90]
> 4. a PDP-11 QBus-to-CheapNet interface which supported both the front-end of a KL10 TENEX/TOPS-20 server and a PDP-11-based IP gateway; these PDP-11 gateways connected to other early IP systems such as the Pluribus IMPs
>
> [The "sexy"] feature was the idea of carrying the packets over a fiber optic link rather than a copper link. This was done using a fiber transmitter/receiver pair from Hewlett-Packard. The salient feature of these devices was that they [could] handle silence on the link. All other transceivers we found at the time would put out noise if there was no transmitter sending on the fiber at a given time. The HP units would

> be quiet in the absence of transmissions. This allowed the CDMA/CD concept to work as if the fiber system was an Ethernet. . . .
>
> [T]he whole concept of fiber optics was new at that time. The idea of a network which could not be tapped by electronic eavesdropping was brand new. This was used to promote BBN's status as a leading-edge network and security house. On more than one occasion, a client from the armed services (particularly the Navy) was brought into a data closet at BBN and shown the CheapNet infrastructure. The demonstrator would begin typing out a long file from one of the timesharing systems onto a local glass TTY. Then he would unscrew the fiber connection from the local hub. He would point out the pretty red light, and then point out that the terminal had stopped printing (not mentioning that all other users on the floor had also lost their network connection). Then he would reconnect the fiber and show that the printing had resumed. This generally impressed the visitor, who was unlikely to be aware of the robustness of the TCP/IP protocols.

SpaceGraph

Starting in 1976, Larry Sher worked on the development and demonstration of SpaceGraph and took part in the quest for applications for it. SpaceGraph provided 3-dimensional displays of data and some images using a vibrating membrane (mounted on a circular metal frame and vibrated at a regular frequency by a loudspeaker) that reflected what was on the screen of a flat display; software fed data to the flat display at just the right time so data that was closer to the viewer in the 3-D display was displayed when the membrane had vibrated forward and data that was farther away from the viewer was displayed when the membrane had vibrated backward in the frame.[91] Although SpaceGraph always excited people who saw it and at one point was licensed to Genisco, SpaceGraph never went very far beyond being a clever device in search of a real use.[92]

21.6 Parallel processors

In the early 1970s, many Honeywell-based ARPANET packet switches and terminal concentrators had been installed and the network was growing.[93] BBN's ARPANET team was looking for something new and innovative to do. Merely maintaining and extending the network was not enough for the ARPANET team of talented and creative engineers. Casting around for what to do next, they got the urge to build a new computer. They spelled out reasons for this urge — the growing network would need more powerful or more reliable packet switches — and sought funding based on these reasons. However,[94] the practical issues of network reliability and growth were not the only issues. The newly available microprocessors were too exciting an opportunity to pass up: The members of the ARPANET engineering team wanted to see if they could build a big machine from a lot of little machines,[95] they wanted to be at the state-of-the-art edge of computer development, and they sought an elegant approach. The urge to design computers was compounded by the hiring of some engineers with even bigger urges and led to a series of multiprocessor designs over the next 20 years,[96] beginning with the Pluribus.[97]

Pluribus

The BBN ARPANET team's (particularly Frank Heart's) emphasis on reliability during the development of its packet-switching system has been widely publicized.[61] Concern for reliability increased in the 1970s as the BBN ARPANET team dealt with actual network

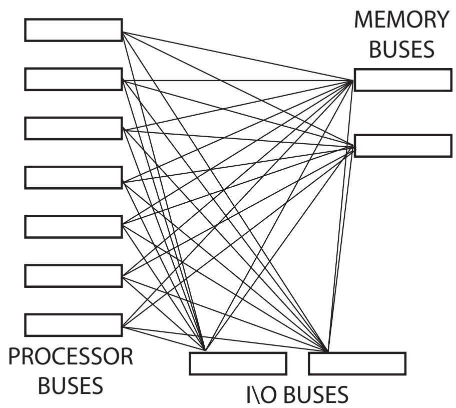


Figure 21.4 Pluribus architecture.

operation including pathological problems. Thus, as the team looked around for a suitable computer on which to build a second-generation packet switch, the focus on reliability had reached mania proportions. This reliability emphasis, combined with the availability of microcomputers, led the team toward a multiprocessor approach for the second-generation packet switch.

There were several specific goals for the new packet-switching system:[98]

- expandability of I/O (more than seven hosts and IMPs)
- modularity; for example, very small IMP for single distant spur host
- expandability of memory; for example, buffering for satellite links or faster links
- reliability; for example, more reliability than Honeywell IMPs, better self-diagnosis, and simplicity of isolating and replacing failing units

Hardware architecture. The team designed a highly modular and redundant multiprocessor system and named it the Pluribus.[99] As shown in Figure 21.4, the Pluribus was based on a modular bus structure and bus-interconnection structure that provided a completely "symmetric" machine in which the processors executed instruction sequences independently of one another and any processor could execute any task in any memory module. Furthermore, machines could be configured with different numbers of buses of various types depending on the needs of the application, and redundancy was supported in the hardware by provision of at least two of each kind of resource on different buses. There was a lot of debate at the time about the best architectural approach to parallel processing; different groups were trying different approaches, and many groups — including BBN — loudly proclaimed why their approach was right and the others were "less right." BBN additionally made its case by implementing its ideas and successfully deploying a variety of experimental and extensive operational applications. The knowledge gained (both positive and negative) helped build a foundation for a next generation of hardware and software at BBN and elsewhere.

Severo Ornstein remembers finding a processor that would fit with the team's goals for the Pluribus:[100] "As we envisioned [the design], the computers [in the multiprocessor] would need to work very closely together, so closely in fact that they would need to share access to a common main memory. . . . As we would need to intervene in [the] processor/memory connection, we needed to have access to it. Our eye was caught by a new machine, the SUE, that was made in unusually modular form by Lockheed Electronics and in which access to this connection was explicitly made externally available."[101,102]

The hardware elements of the Pluribus were:[98]

- processor buses, with one or two SUE processors with up to 8K of 16-bit words of local memory
- memory buses, with 8K memories on each that were sharable by the processors and I/O devices
- I/O buses with shared I/O devices, including
  - ARPANET host interfaces (designed by Marty Thrope)
  - modem interfaces (designed by Ben Barker)
  - satellite interfaces (designed by Randy Rettberg)
  - Pseudo Interrupt Devices (PIDs), aids to task scheduling that went on the I/O buses (conceived by Ben Barker and me with general architecture by Barker and detailed design by Mike Kraley)
  - real time clocks (designed by Kraley)
- bus couplers, used to connect all three kinds of buses to one another and to provide address mapping (conceived by Ben Barker and designed by Tony Michel)
- independent power supply on each bus (a standard SUE component)
- bus arbiter on each that controlled what had access to the bus each cycle (a standard SUE component)

The technology for this system involved processor boards, memory boards, interface boards, and so forth, all plugged into a chassis, with 2.5-inch-wide ribbon cables making the connections among the boards (see Figure 21.5). Systems with a few to 20 or so processors were envisioned. In addition to the shared memories accessible by all processors, each processor board had a memory bank of its own that was faster to access than the common memory.

Multiprocessor programming and reliability software. In a multiprocesser system with many independent tasks to perform (e.g., packet-in, packet-out, routing, etc.), the question of how to schedule tasks arises. The Pluribus approach was to divide the various software components of the overall program into "strips" of code not longer than 100 instructions or so, such that a processor could look for a higher-priority task each time it finished executing a strip. Mike Kraley remembers,

> [Our] important idea here was cooperative multitasking. [T]he contemporary literature was straining to figure out how to do what we now call pre-emptive scheduling — worrying about interrupts, how to save context, when it was safe to interrupt, locking, etc. With our software [for the specific packet-switching application], we were in the rather unique position of being in complete control and having a good understanding of the task. Hence we could "trust" ourselves to break the work up into "strips." We didn't have locks, at least at the hardware level.

In the interests of speed, processors could ask the Pseudo Interrupt Device what was the next priority task that the processor should execute.

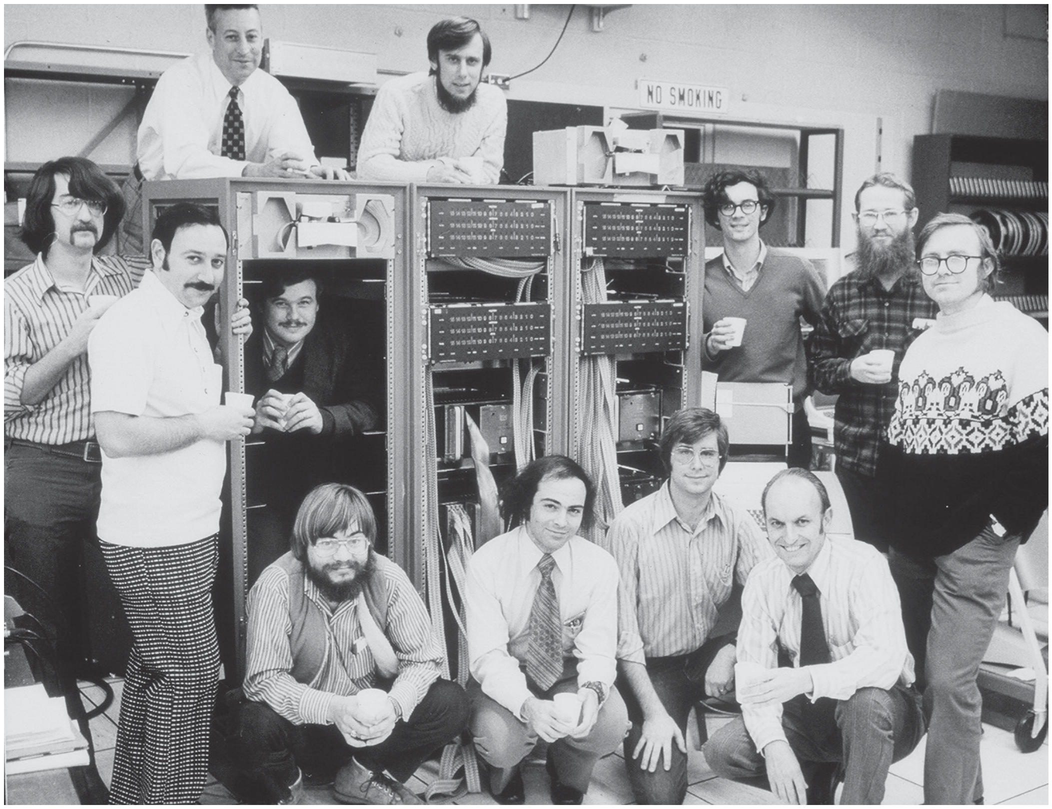


Figure 21.5. A Pluribus multiprocessor and members of the Pluribus team: left to right, Dave Katsuki, Jerry Cherniack, Frank Heart (above rack), Ben Barker (in rack) Tony Michel (squatting), Severo Ornstein (above rack), Marty Thrope (squatting), Bob Bressler (kneeling), Mike Kraley (standing), Dave Francis (squatting), Steve Jeske, Will Crowther. (Photo courtesy of Frank Heart.)

Some of the programmers who worked on the project were Will Crowther, Bernie Cosell, John Robinson, Bob Bressler, and Bill Mann (who had returned to BBN from CCA[103]). The programmers had two major tasks to accomplish: redeveloping the ARPANET packet-switching software for a multiprocessor architecture and making the system fault tolerant.[104]

Randy Rettberg recalls, "Reliability was a fetish in the Pluribus age," and the BBN team worked for years on constructing a fault-tolerant software system for the Pluribus. John Robinson worked on a variety of Pluribus-based projects from 1974 to 1982, and he says, "Through all of this [I was] evolving the 'reliability' software. A good summary is [Katsuki's paper[105]]." Rettberg continues,

> It worked pretty well. I remember that there was no simple way to stop a machine. You had to run a program that shut it down faster than it could fix itself. I also remember that the stage system [the Pluribus subsystem that detected bugs and attempted to recover from them] could recover from a once-a-week bug. But, we eventually had a problem, that Bill Mann worked on, where auto repairing the bug cut throughput to 50 percent. When Bill turned off the automatic repair, he had to fix all the once-a-week bugs to find the culprit.

Applications, productization, and commercialization. The Pluribus hardware architecture and programming methodology were quite flexible, and several application systems were developed for the Pluribus.[106] After the initial High-Speed Modular IMP[107] (HSMIMP) was developed, a TIP[108] version was also built.[109,110] Other Pluribus systems included the Private Line Interface (the first demonstration of end-to-end packet-network security); the CCP (Communications Control Processor);[111] a Very Distant Host converter (to connect remote computers to the network); an innovative message system that was designed but not built, a Pluribus Satellite IMP system;[112] and X.25 extensions of the IMP system for a large commercial bank.

Ultimately, BBN patented the technology[113] and established a significant Pluribus manufacturing capability (that included purchase of the Lockheed SUE factory in Hong Kong) and a broadly distributed on-call and scheduled maintenance capability. The Pluribus became a comprehensively documented,[114] commercial, off-the-shelf multiprocessor computer system that was sold and deployed well into the 1980s.

In time, however, as Pluribus programs grew so big that the programmers could not avoid doing lots of explicit memory management, the machine developed a reputation of being too hard to program. In fact, the software for about half of the later applications of the Pluribus was not written to take advantage of the ability to randomly access any memory from any processor.

Butterfly

In the mid-1970s Will Crowther and his wife divorced, which resulted in Will's thinking about what to do next. At work, we saw him spending a good bit of time on two things: developing the first computer adventure game (Adventure) and drawing "butterfly diagrams." With Pluribus working, the BBN engineers wanted to build bigger and better parallel processors. Mike Kraley says:[115]

> Again, [the Butterfly computer] was fundamentally Willy's idea. He and I spent many blackboard sessions trying to figure out how to interconnect processors and memory using our Pluribus notions of symmetric processors with shared memory. Our n x m approach of bus couplers [as in the Pluribus] didn't scale well. So the idea of the butterfly switch was born [see Figure 21.6]. I recall doing some software simulations — we were concerned about deadlock in the switch. This was the last thing Will did before moving west.

In May 1976, Crowther moved to Xerox PARC in California, following in the footsteps of Severo Ornstein.

Even with Crowther gone, however, the concept of building a parallel processor based on a butterfly switch was launched and would not stop. Over a period of years Will's butterfly drawings turned into a real computer (as shown in Table 21.1), primarily funded by ARPA.

Butterfly architecture. Figure 21.6 shows a basic Butterfly switch with three ranks of 2x2 switching nodes connecting eight processors to eight memories. The figure shows how processor 1 and processor 7 both address memory 101 (5 in binary) to reach memory 5. By using bigger switching nodes (e.g., a 4-way switch as in Figure 21.7),[116] a Butterfly switch can connect more processors and memories. It is also possible to use more ranks of switching nodes. Thus, the Butterfly was another shared-memory machine (like the Pluribus) in which all of the processors could get to all of the shared memory. However, it was built with later generations of technology for electronics integration that enabled more processors (up to hundreds) to share a common memory. The Butterfly used the popular 68000 series processors from Motorola.

Table 21.1 Butterfly time line. (Courtesy of Mike Kraley and John Goodhue.)

April 1975 First proposal to ARPA under advanced memory concepts.

December 1975 Second proposal submitted to ARPA for "computer architecture issues for real-time systems."

April 1976 Patent disclosure.

May 1976 Crowther leaves; Wes Clark and Mike Kraley try to write down all the architecture ideas.

August 1976 TENEX simulation of Butterfly switch.

September 1976 Discovery of deadlocks in Butterfly networks.

October 1976 Retreat-and-discard strategy for Butterfly networks.

December 1976 Brief Fort Meade (government customer) on Butterfly.

February 1978 International Solid State Circuits Conference in San Francisco, where Kraley and Rettberg decided they would pressure their BBN division to build both the MBB and the Butterfly.

June 1981 Two Butterfly processors talk via switch.

August 1981 Butterfly VLSI switch design starts.

November 1981 Ten-processor Butterfly built.

January 1982 Chrysalis operating system runs on 10 processors

March 1982 FIFO chip back from MOSIS; Buttefly chip submitted.

March 1982 Voice Funnels at ISI and Lincoln Labs carry a packet voice telephone call across the wideband net.

May 1985 Benchmark results from a 256-processor Butterfly.

October 1985 Butterfly Satellite IMPs (BSATS) replace Pluribus Satellite IMPs in the DARPA Wideband Network.

1986 BBN Advanced Computer Inc. starts to capitalize commercially on BBN's parallel processing technology.

August 1989 BBN ACI launches the TC-2000 (Butterfly II).

June 1990 TC-2000 ADA Compiler ships.

March 1991 Lawrence Livermore Laboratory publishes "Attack of the Killer Micros," a 200-page report on application development and benchmarking.

August 1991 BBN ACI closes down. Remaining parallel-processing activities folded back into BBN Laboratories.

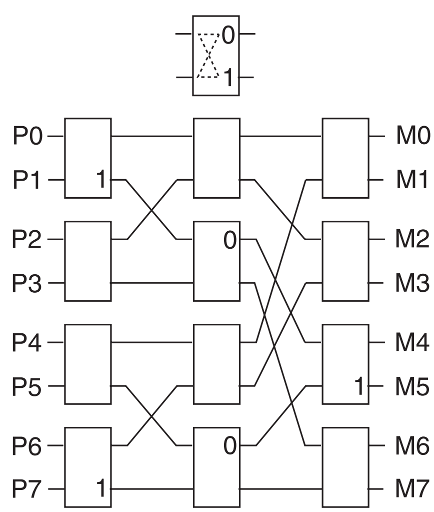


Figure 21.6 Simple example of a Butterfly switch.

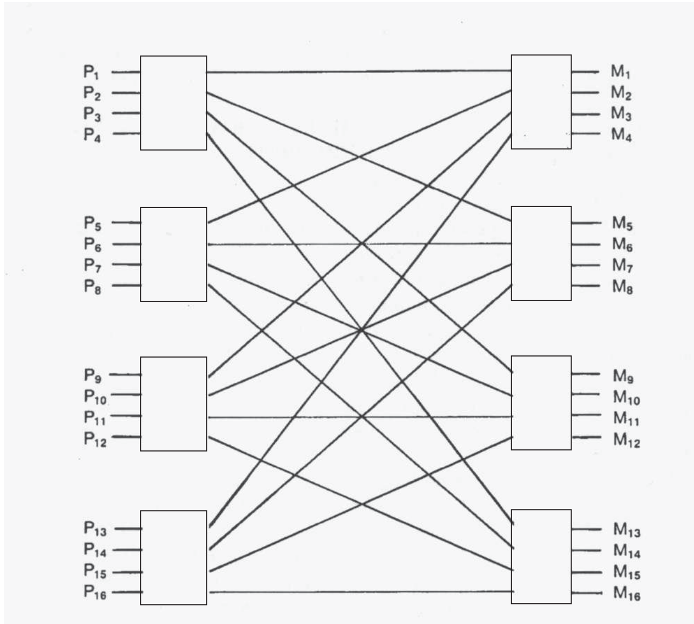


Figure 21.7 Butterfly switch using 4-way switching nodes.

Randy Rettberg says,

> The real innovation in the butterfly switch was the packet-switching nature of the switching where a header in the packet determined the path through the switch.[117]

A packet of data (e.g., when a processor wants to store a word of data in memory) consists of the address of a memory location and the data word. The first bits of the address specify a particular memory module, and they are used serially, one by one, to switch the rest of the packet through the Butterfly switch (look again at Figure 21.6). Rettberg continues,

> There was lots of discussion about whether this was circuit-switching but without an external controller, or if it was packet switching but without packet storage in the switch. It was obviously something new. . . .
>
> Before we got the Butterfly contract, we did thorough designs and simulations. Bernie [Cosell] did the simulations. We actually solved the "hot spot" problem by using retreating and discard in the Butterfly switch, but we kept it secret. As a result, nobody believed us.

As John Goodhue remembers,

> Phil Carvey joined BBN at the beginning of the Butterfly project, and was the lead hardware designer of the first processor and switch nodes. Ward Harriman later joined the project (and eventually became the lead hardware designer when Carvey went on to another project) doing the package and a higher performance processor card. I joined the project shortly after Harriman. My assignment was to write the Voice Funnel, the first application to run on the Butterfly.[118] Bill Mann joined the project shortly after I did and wrote the Chrysalis operating system. There were others on the original project, but the combination of Rettberg, Harriman, Carvey, and Mann stand out as the key figures at that time.

While the first Butterfly computers were being built, ARPA was setting up the MOSIS[119] facility to fabricate VLSI[120] designs from small engineering groups around the country. Naturally, BBN became a hotbed of interest in VLSI, and the Butterfly hardware team built a VLSI version of the Butterfly switch. Rettberg remembers John Goodhue, Mike Kraley, Phil Carvey, and Paul Bassett working on the VLSI design. The higher packaging density of the VLSI Butterfly switch chip enabled a threefold increase in the number of processors per rack, making it possible to build a 256-processor machine.

Multiprocessor programming. As mentioned above, Bill Mann wrote the original Butterfly operating system, Chrysalis, with some design contributions from Rettberg. At some point Will Crowther returned to BBN from Xerox PARC, and Bob Thomas moved from the distributed systems group he started at BBN to the Butterfly team.[121]

John Goodhue recounts,

> Like the Pluribus, the Butterfly retained the notion of logically shared but physically distributed memory, so that there was faster-access "local" memory that appeared as slower-access "remote memory" to all of the other CPUs in the machine. Two styles of programming evolved: a cooperating sequential processes model where programmers took special care to keep all relevant data in local memory, and [Crowther's] "uniform system" model[122] where most data structures were scattered across the memory without regard for the local/remote distinction. The latter idea was developed by Crowther when he returned to BBN from PARC, and was an early seed for his thinking on Monarch.[123] Bob Thomas built the tools to support it.

Applications. The Butterfly computer was used both for another generation of networking applications and for more general computational work. The voice funnel application has already been noted. John Robinson also comments,

> The wideband net[124] was based on Butterfly (the Butterfly Satellite IMP, or BSAT[125]). Winston Edmond and Walter Milliken did much of the software work. It built on the Pluribus satellite work of Dick Binder, which in turn evolved out of 316 satellite IMP work of Randy Rettberg, Nils Liaaen, and later Winston Edmond.

Bob Hinden, Eric Rosen, Linda Seamonson, and others built an IP gateway based on the Butterfly.[126]

On the more general computation side of things, Bob Thomas, Will Crowther, and others (including researchers at the University of Rochester) were doing experiments with applications of common mathematical methods such as Gaussian elimination and finite-element analysis. The goal was to see what degree of efficiency could be obtained with larger numbers of processors. In 1985 the University of Rochester had bought a 128-processor Butterfly and DARPA had bought 10 16-processor machines. Before they shipped, the machines were combined into a 256-processor system with excellent benchmark performance results.[116,127] Also during this time, Jeff Deutsch, a graduate student from U.C. Berkeley, spent the summer at BBN developing a parallelized version of the SPICE circuit simulator on the Butterfly.

Commercialization of the Butterfly. In 1986 BBN started BBN Advanced Computer Inc. (ACI), a subsidiary to develop Butterfly-based high-performance computers. The financing of this activity is described in Chapter 6. Paul Castleman and Chan Russell (who had been at BBN Software Products Corporation) joined BBN ACI as president and VP for software. Randy Rettberg transferred from the R&D part of BBN to be VP for hardware and brought with him the Butterfly technology plus the engineers and programmers who had been part of the Butterfly project in the R&D part of BBN. Additional engineers and programmers were hired by BBN ACI.

While BBN ACI sold and serviced Butterfly systems (the first three items in Table 21.2), its business plan relied on a new machine called the TC-2000 (the fourth item in Table 21.2), and that was the primary focus of BBN ACI development activity.

Table 21.2 Versions over time of the Butterfly computer.

Butterfly: The machine that DARPA originally funded before BBN ACI was formed. The Butterfly ran the Chrysalis operating system and evolved through several hardware generations. The final version used the VLSI switch chip and a Motorola 68020 CPU, and had both Multibus and VMEbus I/O.

Butterfly Plus: The product name given to the Butterfly by BBN ACI. BBN ACI sold and supported the Butterfly Plus for DARPA programs that wanted Chrysalis machines (e.g., Butterfly Gateways, BSATs, and some early Strategic Computing Initiative programs). It was not actively marketed.

GP-1000: The same hardware as the Butterfly Plus, but it ran UNIX instead of Chrysalis. The GP-1000 was the first product to be actively marketed by BBN ACI.

TC-2000 (also called the Butterfly II): A next-generation Butterfly that had (a) new hardware from the ground up (new package, new switch, new processor nodes based on Motorola's 88000); (b) a port of the UNIX operating system that was first deployed on the GP-1000; and (c) a pSOS "bare-bones" real-time operating system that one could run on a subset of the nodes in the machine.

The TC-2000 was conceptually similar to earlier Butterfly systems, but it had all new hardware and a far more powerful suite of software tools. The main characteristics of the machine were driven by the real-time simulation applications that were the primary business focus of BBN ACI:

- A combined UNIX and pSOS operating environment; programmers ran development tools and complex software under UNIX, and real-time applications such as data collection in the "bare bones" pSOS environment.
- The GIST performance analyzer and the TotalView debugger for debugging and tuning multiprocessor software in the combined UNIX/pSOS environment.
- A port of the Telesoft ADA compiler, which was required for government simulation applications at the time.
- A return to the emphasis on reliability that had been present in the Pluribus but less prominent in the Butterfly, including redundant switch, power, and I/O.
- VMEbus I/O on every processor, allowing attachment of specialized third-party hardware.
- Use of the Motorola 88000 RISC processor to gain the floating point performance that had been lacking on previous microprocessors.

Regarding the TotalView debugger, Bob Thomas remembers,

> The TotalView debugger was inspired by the Jericho Pascal debugger (which was developed by Ray Tomlinson with kibitzing from Harry [Forsdick] and me). TotalView was initially contracted out to Think Technologies and Steve Lawrence, who had many good ideas himself. BBN ACI later hired Steve who remained the TotalView developer.

Googling on "TotalView the parallel program debugger" shows that this program has been ported to various other parallel machines, long outliving the Butterfly platform.

Outside of the main sales efforts in real-time simulation, the TC-2000 also attracted interest in two other areas:

- *High-transaction-rate databases*: As a proof of concept, BBN programmers Robert Wells, Dave Barach, and Bob Goguen (with help from some people from Oracle) parallelized the Oracle database software to run on hundreds of processors. John Goodhue describes this as a tour de force of programming. However, the price performance of the hardware (which was designed for lots of general purpose I/O devices, not lots of disks), was less than compelling. Also, Oracle's founder and CEO Larry Ellison had recently invested in a rival parallel-processing company.

  High-performance computing: Debbie Fanton and Bill Celmaster doggedly pursued the security and national laboratory communities. Their one significant success was a contract from the Lawrence Livermore National Laboratory, which enabled a second life for BBN ACI.

The TC-2000 was launched three years after the formation of BBN ACI. One year later, sales efforts had not lived up to expectations. The demise of Strategic Defense Initiative and decline in defense spending reduced demand for high-performance real-time computing. Also, by the time the TC-2000 was ready to ship, hardware prices and the power of single-processor computers was such that the parallel-processor real-time market didn't really develop. Finally, BBN got caught in an intra-ARPA power struggle that caused potential Butterfly purchases in the DoD community to be seriously delayed or stopped altogether.

A second attempt. When the R&D funding ran out,[128] Paul Castleman and Chan Russell left BBN ACI, and Randy Rettberg moved back to the R&D side of BBN. Ben Barker then took a turn as leader of a much-downsized BBN ACI, with John Goodhue and Tom Downey running the engineering and marketing groups. Over the next year, BBN ACI tried to re-aim its business at the high-performance computing market. John Goodhue recalls,

> Bob Thomas, Rich Schaaf, and their software teams had created a UNIX operating system and development tool suite that attracted interest from many of the government high performance computing labs. Julie Tiao led a hardware project that increased the maximum size of the machine from 64 to 256 processors, enabling sufficient high performance computing sales to cover our operating costs for a year. Tom Blackadar ran a 2-person customer service operation. Our largest customer was Livermore Labs — by the end of the year they had found two applications that outperformed their Cray systems on TC-2000s that cost far less. These ran in their production facility for many years.
>
> At the same time Guy Fedorkow, Dan Tappan, and others came up with a hardware design that would run the same software with competitive price performance in the high performance computing marketplace. That meant jettisoning features like the integrated VMEbus I/O, accommodating higher processor counts (prospective customers were asking for 1,000 processors), and incorporating the next generation of RISC processors. The new design got positive reviews from the prospective customers. However, it would have required another $20M to prototype and launch the new machine, which was beyond what BBN could reasonably invest. Levy and Barker tried to sell it to Cray, which was considering an investment in large scale parallelism. Cray did purchase licenses for TotalView and Gist, but decided to pursue a more evolutionary hardware path with fewer CPUs, liquid cooling, and vector processing. With no path to a commercially viable product, BBN ACI closed its doors in 1991. Most of the engineering team moved to the Emerald project and subsequent Lightstream venture.

Monarch, Monet, Emerald, and Lightstream

Despite the shutdown of BBN ACI, BBN continued pushing various parallel-processing ideas.

Monarch. Somewhat after the Butterfly effort started, BBN's parallel-processing engineers and programmers began thinking about a bigger yet parallel processor. Pluribus had supported a few tens of processors; Butterfly could have a few hundreds of processors; Monarch was to support thousands of processors.[129] John Goodhue remembers,

> Randy [Rettberg], Will [Crowther], and Lance Glasser were the ones that developed the original ideas for Monarch. Phil Carvey and Ray Tomlinson also joined the project early on and added numerous ideas that turned concepts into a buildable machine architecture. Randy was the leader of the Monarch project until he handed it off to me when he became hardware engineering VP at BBN ACI.

Monarch was to be a new generation of computer using the Butterfly switch. Randy Rettberg says,

> The key to the Monarch was that the processor was custom designed for the switch interconnect — everything matched. . . . BBN's focus on high speed signaling made the Monarch possible. We targeted 100 Mbps per pin in CMOS, but hit over 400 Mbps. . . . The modern techniques of dynamic deskewing, low voltage swing signaling and even on-chip termination were pioneered in the Monarch switch. . . .
>
> One of the most important innovations in the Monarch was the "steal" instruction, stimulated by parallel processor LISP work for the Butterfly.[130] The Monarch software could Read, Write, or Steal the contents of a memory location. If stolen, subsequent attempts to read that location were blocked until it was written into. The switch was designed so that simultaneous references to the same memory location would be combined within the switch producing a single memory reference with the result delivered to all requesting processors. The switch had two simultaneous paths so that thousands of reads would not interfere with one write. The steal instruction was the only synchronization mechanism in the machine, but it allowed very fine-grained locking wherever needed in a data structure.

For the Monarch design, Randy Rettberg served as internal entrepreneur, project leader, and architect. Phil Carvey (who always lusted to design the hardware for the most powerful machine in the world) participated, as did Will Crowther (who always had another clever architecture or software idea) and Ray Tomlinson (of e-mail fame, who was equally facile with hardware and software development). Other outstanding programmers and engineers joined the team.

The Monarch project was originally funded by ARPA as a four-year program with the goal of building a 1,000-processor machine. When the Butterfly effort moved to BBN ACI, the Monarch effort went along. Having this big research project side by side with the effort to commercialize the Butterfly probably didn't help the latter. When Rettberg returned from BBN ACI to the R&D part of BBN, the Monarch project came with him. After Rettberg himself left BBN, Ben Barker (then in BBN's corporate development office) took over oversight of the Monarch project, with me helping him with interactions with our government customers.

Within a year or two, however, the team was thinking about an even bigger machine, which was proposed to the government customer at Fort Meade. This was to have 64,000 processors, 128 gigaflops, and 64 trillion memory references per second, all in a machine that would fit in a 40-by-40-foot computer room. An intense design study was done and showed that it would be possible for software to take advantage of such massive parallelism.

However, a 64K processor machine design (not to mention the commercial goals of BBN ACI) distracted from the Monarch implementation. At one point the BBN ACI people were working on five generations of parallel machines: Butterfly, Butterfly Plus, GP-1000, TC-2000, and Monarch. Eventually the Monarch project was well behind schedule for finishing the 1,000-processor system. Naturally, this caused trouble with the ARPA sponsor, where the project was already being touched by the ARPA politics that had affected the Butterfly. The ARPA politics, the slip in building the 1,000-processor Monarch, and disbelief by some in the government of BBN's design claims for the even bigger machine eventually led both projects to be abandoned.

Monet, Emerald, and Lightstream. Out of Monarch came the Monet (Monarch Network) switch. This used Butterfly switch routing but included low-level routing in the switch. BBN started an internal R&D project for the Monet switch in 1989–1990, and Ben Barker and John Robinson tried to sell this idea to Fort Meade and to Vint Cerf at CNRI. Barker says,[131]

> The concept was to use the self-routing serial Monarch switch chips to form a high-speed self-routing local or wide-area network. [The potential] clients were delighted at the fact that the hardware affixed the source route at the end of the packet, making it hardware-provable where the packet came from. They were also pretty excited about a cheap 400 megabit switch.

At this same time, BBN Communications Corporation, which had had great success with its packet switches but missed the router generation, was struggling to get in front of the next market and technology wave. Jack Holloway (a founder of the Symbolics LISP machine company) had joined BBN, and he, Bob Hinden, John Goodhue, and others pushed the idea of using BBN's multiprocessor ideas to build an ATM (asynchronous transfer mode) switch. ATM switches looked like they might be the next big thing in networking. Thus, as BBN ACI was being shut down, the engineering team moved to BBN Communications and began work on what was called the Emerald project, which took ideas from the Butterfly, Monarch, and Monet.

Work began in BBNCC on Emerald; and, after months of negotiations, a deal was done with Ungermann Bass (UB, then an important local area networking company) for UB to provide the LAN interfaces and other requirements to make a complete switch. Much excellent design and development was done on the new ATM switch. Nonetheless, BBN Communications Corporation's business volume continued to erode, and eventually, Ben Barker and I were sent to BBNCC to try to save things. Barker (who had been called back from a yearlong sabbatical he was taking, sailing to Europe and back to recover from his BBN ACI struggles) took responsibility for the Emerald project and work with UB. I, in turn, took responsibility for BBNCC's traditional network systems business which was eventually folded back into BBN's R&D organization. As BBNCC was being phased out, BBN and UB did a joint venture to finish and market the ATM system. Ben Barker and the Emerald team left BBN to participate in the new joint venture, which was named Lightstream. Jonathan Crane was brought in to replace Barker as leader of the joint venture.

John Goodhue notes that Ben Barker was a central force behind both the reconstitution of BBN ACI and the formation of Lightstream:

> There were quite a few times where we would have given up if Ben had not been such an optimist and persistent advocate.

About a year later, Cisco bought the Lightstream company from BBN and UB, and the onetime members of the BBN parallel-processing team went along to Cisco. Nonetheless, BBN continued to be active in parallel processing in Steve Blumenthal's group, where the Monarch project and some of the Butterfly and Monarch engineers ended up. As is noted in the section on routers in Chapter 17,

> a BBN team led by Craig Partridge, Josh Seeger, Walter Milliken and Phil Carvey designed and built a prototype of the world's first 50-gigabit-per-second router. . . . Variants of this router architecture are now the standard in the router industry, and BBN's paper on the router[132] is required reading at many corporations.

Acknowledgments

In addition to everyone who is noted as a source section-by-section and in the note and references, Alex McKenzie, Ray Nickerson, and John Swets provided review and additional input. I regret that space limitations prevent mentioning by name many other BBN people who also worked on the projects mentioned in this chapter.

Notes and References

1. John McCarthy et al., *LISP 1.5 Programmer's Manual*, MIT Press, Cambridge, MA, 1962; second printing 1965.

2. Sources of information for this section were Lyn Bates (e-mail of July 6, 2003); Danny Bobrow (e-mails of July 26, 2002, and August 5, 2003); Jim Goodwin (e-mails of May 27, 2010); Alice Hartley (e-mail of July 7, 2003); Dan Murphy (e-mail of October 25, 2003); Bruce Roberts (e-mail of July 7, 2003); Smokey Wallace (e-mail of September 24, 2002); and John Vittal (e-mails of July 29, 2003, and May 26–27, 2010).

3. Another source for this section was Steele and Garbriel's evolution-of-LISP paper: Guy L. Steele Jr. and Richard P. Gabriel, "The Evolution of LISP," *ACM SIGPLAN Notices*, 28(3):231–270, March 1993. A version of this paper was later published on pp. 233–308 of *History of Programming Languages — II*, Thomas J. Bergin Jr. and Richard G. Gibson Jr., eds., ACMPress, NY, and Addison-Wesley, Reading, MA, 1996; this later volume also included ancillary material related to the paper and its public presentation (pp. 309–330).

This paper includes discussion of how LISP is different from other computer languages and why it is so useful for certain kinds of programming.

4. Steven Levy, *Hackers: Heroes of the Computer Revolution* Anchor Books, New York, 1984. The author of this book is a technology writer from California, not the longtime BBN executive, Stephen Levy, who wrote Chapter 6 of this volume.

5. John McCarthy, "History of LISP," *ACM SIGPLAN Notices*, 13(8):217–223, August 1978.

6. L. Peter Deutsch and Edmund C. Berkeley, "The LISP Implementation for the PDP-1 Computer," in Edmund C. Berkeley and Daniel G. Bobrow, eds., *The Programming Language LISP: Its Operation and Application*, MIT Press, Cambridge, MA, 1963.

Deutsch's LISP was the first interactive LISP and influenced the development of LISP on the PDP-6 at MIT (and thus MAC LISP).

7. D. G. Bobrow, L. Darley, and D. Murphy, The BBN-LISP System, BBN Report 1346, February 1, 1966.

8. D. Bobrow and D. Murphy, "Structure of a LISP System Using Two-Level Storage," *Communications of the ACM*, 10(3):155–159, March 1967.

9. Dan Murphy, "Origins and Development of TOPS-20." Available at `tenex.opost.com`, 1989 and 1996.

10. D. Bobrow and D. Murphy, "A Note on the Efficiency of a LISP Computation in a Paged Machine," *Communications of the ACM*, 11(8):558–560, August 1968.

11. D. G. Bobrow, L. Darley, and L. P. Deutsch, The BBN 940 LISP System, BBN Report 1539, July 1, 1967; D. Bobrow, D. L. Murphy, and W. Teitelman, The BBN 940 LISP System, BBN Report 1677, April 1, 1968.

12. W. Teitelman, Design and Implementation of FLIP, a LISP Format Directed List Processor, BBN Report 1495, July 1, 1967; D. G. Bobrow and W. Teitelman, Format-Directed List Processing in LISP, BBN Report 1366, April 1, 1966

13. Warren Teitelman, "Automated programming: The Programmer's Assistant," in *AFIPS Proceedings, Fall Joint Computer Conference*, vol. 41, pp. 917–921, Montvale, NJ, AFIPS Press, 1972.

14. In addition to his LISP and AI work, Teitelman was in demand for other projects. For instance, he helped bail out a BBN project for the Pacific Coast Stock Exchange that was in trouble (quickly writing code to decode cryptic ticker-tape information).

Teitelman also pulled together a network simulation and display program for Bob Kahn before BBN had won the ARPANET contract: W. Teitelman and R. E. Kahn, "Network Simulation and Display Program," in *Proceedings 3rd Annual Princeton Conference on Information Sciences and Systems*, p. 29, Princeton University, Department of Electrical Engineering, Princeton, NJ, March 1969.

This 1969 Teitelman and Kahn paper about the network simulation system was BBN's first published paper relating to the Internet. (Once the ARPANET contract was won, Frank Heart apparently discouraged Kahn from using the simulation software to anticipate problems with the algorithms Crowther, Cosell, and I were coding.)

15. Warren Teitelman, "PILOT: A Step toward Man-Computer Symbiosis," PhD thesis, MIT, September 1966; Technical Report MAC-TR-32, MIT Project MAC.

16. H. Rudloe, "Tape Editor," Bolt Beranek and Newman Inc., Cambridge, MA, January 1962; Program Write-up BBN-101.

17. W. Teitelman, D. G. Bobrow, A. K. Hartley, and D. L. Murphy, *BBN-LISP: TENEX Reference Manual*, Bolt Beranek and Newman Inc., Cambridge, MA, 1971.

18. The Common LISP development in the 1980s partially was aimed at reducing the number of LISP variations.

19. Bobrow also remained active in coming up with improvements relevant to LISP, such as spaghetti stacks and block compiling: Daniel G. Bobrow and Ben Wegbreit, "A Model and Stack Implementation of Multiple Environments," *Communications of the ACM*, 16(10):591–603, October 1973.

20. For readers who remember Jim Goodwin leaving BBN, going to Europe, and then settling in Sweden, he says that he used the 360-based LISP system in Sweden and did not participate in the porting.

21. John Vittal notes, "There was the LISP language (and indeed, the InterLISP language, too), and then there was the InterLISP programming environment (written in InterLISP). This was one of the first, if not the first, instance of what is referred today as an IDE (Integrated Development Environment). For example, editors, compilers, etc., while callable from a program, could be considered part of the environment instead. Not even MIT LISP, at the time, had as extensive a set of tools and seamless integration as BBN-LISP."

22. Warren Teitelman and Larry Masinter, "The InterLISP Programming Environment," IEEE *Computer*, April 1981, pp. 25–33.

23. BBN's participation continued well into the late 1970s and perhaps beyond. For instance, Alice Hartley maintained the interpreter, garbage collector, hand-coded machine language functions, compiler, and spaghetti stacks; Daryl Lewis maintained the terminal, user, and TENEX interfaces of InterLISP; and Jim Goodwin wrote speedy library routines for LISP in LAP (LISP Assembly Program) and LISP virtual memory routines in MACRO-10 assembly language. John Vittal remembers that some of this maintenance was done without explicit funding. BBN people (e.g., John Vittal and Martin Yonke) also did some additions to the LISP environment (in the form of packages).

Some relevant references are: W.R. Sutherland, A.K. Hartley, and D. Lewis, Natural Communication with Computers: Final Report, Vol. V, INTERLISP Development and Automatic Programming Oct 1970–Dec 1974, BBN Report 2976 (Vol. V), December 1, 1974; M.C. Grignetti, R. Bobrow, and A. Hartley, Interlisp Performance Measurements, BBN Report 3331, June 1, 1976; A.K. Hartley, Interlisp-11, BBN Report 4076, March 1, 1979; A.K. Hartley, INTERLISP Maintenance: Final Report, BBN Report 4304, February 1, 1980; M. Bates, R. Bobrow, and B. Wagreich, Interlisp Primer, BBN Report 4353, March 1, 1980.

24. Warren Teitelman et al., *InterLISP Reference Manual*, Xerox Palo Alto Research Center, Palo Alto, CA, first revised edition, 1974.

25. A Web search reveals many instances of the following quote ("Interlisp Programming Manual," W. Teitelman, TR, Xerox Rec Ctr 1975):

"[Interlisp is] a dialect of Lisp developed in 1967 by Bolt Beranek and Newman (Cambridge, MA) as a descendant of BBN-Lisp. It emphasizes user interfaces. It is currently supported by Xerox PARC.

"Interlisp was once one of two main branches of LISP (the other being MACLISP). In 1981 Common LISP was begun in an effort to combine the best features of both. Interlisp includes a Lisp programming environment. It is dynamically scoped. NLAMBDA functions do not evaluate their arguments. Any function could be called with optional arguments."

26. John Vittal sees a connection between BBN sidelining itself and the disappearance in funding to BBN for LISP development resulting from efforts elsewhere to commercial LISP (at Gold Hill, LISP Machines, Inc., and other places mentioned below).

27. Warren Teitelman himself has written about the history of InterLISP: Warren Teitelman, "History of InterLISP," *LISP50: Celebrating the 50th Anniversary of Lisp*, C. Herzeel, ed., ACM, New York, 2008, pp. a5.

28. According to John Vittal, there were two ports of InterLISP to the Alto; in one instance, the whole of InterLISP was put on the Alto; in the other instance, a graphical front end to InterLISP was put on the Alto that interacted with the main body of LISP on TENEX.

29. See page 544 for mention of BBN's work on parallel-processor versions of LISP.

30. I believe a copy of that position paper is available in the following: David Walden, "An Informal Note and Some Readings on Extensible Languages," Bolt Beranek and Newman Inc., Cambridge, MA, 1968, BBN accession number 001.642 W162I, 389 pages.

31. At the time I was very interested in programming languages and macro processors. In particular, I was working toward a master's degree in computer science at MIT and, having finished the course work, was supposed to be developing and writing a thesis on a sophisticated macro processor front end for a high-level language that future Internet legend Dave Clark was developing for his master's thesis (we both had MAD codeveloper Bob Graham, who was at that point deeply involved in the Multics development, as our thesis supervisor).

32. I've always claimed that my glibness about extensible languages and doing an implementation that didn't work made me one of the fathers of the PROPHET system. I also never finished my thesis work and never got the MIT M.S. for which I had done all the course work; rather, I got involved in BBN's ARPANET proposal and then in the ARPANET development.

33. E-mail of June 3, 2003.

34. C.R. Morgan and A. Evans, Communications Oriented Language (COL): Language Implementation and Usage, BBN Report 3533, May 1, 1977; C.R. Morgan and A. Evans, Communications Oriented Language (COL): Language Definition, BBN Report 3534, May 1, 1977; A. Evans and C.R. Morgan, Analysis of Preliminary Designs for a Common Programming Language for the Department of Defense, BBN Report 3775, March 1, 1978; J. Walker, A. Evans, C.R. Morgan, J.G. Wood, and M. Zarnstorff, PRAXIS Language Reference Manual, BBN Report 4582, July 1, 1981; A. Evans Jr., A Comparison of Programming Languages: ADA, PRAXIS, PASCAL, C, BBN Report 4634, May 1, 1981.

35. E-mail of June 3, 2003.

36. Art Evans had been on the faculty at MIT, where he was Bob Thomas's thesis supervisor. When Art wanted to leave MIT for industry, Bob asked my division to interview him, and we hired him.

37. In addition to the references cited in this section, information for this section came from Danny Bobrow (e-mails of August 26, 2002, and June 4, 2003), Bob Clements (e-mail of January 25, 2004); Ted Strollo (e-mails of October 28 and July 16, 2002, and July 19 and 21, 2003), and Ray Tomlinson (e-mails of June 4 and July 7, 2003.

38. B. Lampson, M. Pirtle, and W. Lichtenberger, "A User Machine in a Time-Sharing System," *Proceedings of the IEEE*, 54(12):1766–1774, December 1966.

39. The PDP-1b and -1d developments mentioned in this paragraph are described in Chapter 4.

40. Ted Strollo joined BBN full time in 1965; his master's thesis advisor introduced him to Jerry Elkind, who recruited him to BBN. (Before that, he had had a summer job at BBN in 1962 while at MIT.) While he started at BBN doing control systems work with Elkind and Jerry Burchfiel, somehow Strollo moved into the world of time sharing. In time he had a dual role: He managed BBN's Research Computer Center, originally based on the PDP-1b, and later he also managed the time-sharing research and development department, reporting to Danny Bobrow.

41. A custom disk controller allowed access by both the SDS 940 and the TENEX system described below.

42. Tomlinson, who arrived at BBN from Ken Stevens's speech-processing group at MIT, was involved in numerous projects over the years that are mentioned in this chapter and the companion chapters, and became legendary within (and outside) BBN.

43. D.L. Murphy and R.S. Tomlinson, *BSYS: The BBN SDS-940 Disc Backup System*, Bolt Beranek and Newman Inc., Cambridge, MA, April 1, 1969.

44. Later, Lampson et al. and Bobrow and Strollo et al. were together at Xerox PARC.

45. Burchfiel had come to BBN with a PhD in control theory. Later, when Ray Nickerson was division director of the activities previously in Elkind's and Bobrow's domain, Burchfiel helped Nickerson manage the division, managed the Research Computer Center, and led the packet-radio system development (Chapter 17), among other activities.

46. After leaving BBN, John Barnaby moved to California, became known as Rob Barnaby, and had an influence on the fledgling personal computer industry. The IMSAI web site says, "Fabled madman programmer Rob Barnaby ... wrote the code enabling CP/M to become the first commercial operating system available for personal computers, eventually evolving into IMSAI IMDOS. He also wrote WordStar (at one time the most popular word processing program in the world) for Seymour Rubinstein when he joined MicroPro International." At BBN, Barnaby worked with MRUNOFF, which looks like it might have had some influence on his ideas for WordStar.

47. Ed Fiala joined Lampson, Deutsch, et al. at Berkeley Computer Corporation and went on with them to Xerox PARC when BCC failed.

48. Bill Plummer was coauthor of an influential paper on the MIT PDP-1 time-sharing system: W. Ackerman and W. Plummer, "An Implementation of a Multi-Processing Computer System," *Proceedings of the ACM Symposium on Operating System Principles*, Gatlinsburg, TN, October 1–4, 1967, paper D-3.

49. Nonetheless, some of the Multics ideas influenced the TENEX design.

50. Daniel G. Bobrow, Jerry D. Burchfiel, Daniel L. Murphy, and Raymond S. Tomlinson, "TENEX, A Paged Time Sharing System for the PDP-10," in *Third Symposium on Operating System Principles*, pages 1–10, 1971.

51. "Paged virtual address space, multiple process capability with full provision for protection and sharing, and file system integrated into the virtual address space, building on a multi-level symbolic directory structure with protection, and providing consistent access to all external I/O devices and data streams."

52. Daniel G. Bobrow, Jerry D. Burchfiel, Daniel L. Murphy, and Raymond S. Tomlinson, "TENEX, A Paged Time Sharing System for the PDP-10," *Communications of the ACM*, 15(3):135–143, March 1972.

53. Daniel L. Murphy, "Storage Organization and Management in TENEX," in *Proceedings of AFIPS Fall Joint Computer Conference*, vol. 41, part 1, pp. 23–32, Montvale, NJ, AFIPS Press, 1972.

54. About the time TENEX was originally being finished, Strollo moved to Xerox PARC; however, a year later he was back at BBN, picking up where he left off and continuing for another five years.

55. I see TENEX as third in an evolutionary line of time-sharing developments — PDP-1s at BBN and MIT, SDS 940, and TENEX — each building on the idea of the preceding. CTSS at MIT also probably had some influence. Outside the time-sharing community (which can be thought of as the community of ARPA researchers), many commercial vendors mostly ignored (and even disparaged) time sharing.

56. Ken Thompson, "Reflections on Trusting Trust," *Communication of the ACM*, 27(8):761–763, August 1984. His Turing Award Lecture.

57. Ted Strollo remembers that Xerox would not let its researchers at Xerox PARC buy a PDP-10, so the researchers built an emulator called MAXC that ran TENEX.

58. Remember that most BBN research was done under contract to an outside party, most often the U.S. Government.

59. Around the same time, the question was arising of how to account for use of the ARPANET, the precursor of the Internet. With advice from its contractors, including BBN, ARPA decided to charge for ARPANET use by selling network connections of various capacities for fixed-prices per month or year. A goal was to encourage network use in those beginning Internet days, not to account for it in a way that led users to avoid using the Internet. With the fixed price-for-fixed-capacity system, users would know the cost up front and typically put it in the overhead cost pool and forget about it. These examples of time-sharing and ARPANET accounting in the early 1970s anticipated the pressures we see today for the phone companies, Internet service providers, cable companies, and so on, to provide fixed price offerings with predetermined costs rather than minute- (and perhaps distance-) based accounting with uncertain costs.

60. Severo Ornstein, e-mails of April 20, 2010

61. Katie Hafner and Matthew Lyon, *Where Wizards Stay Up Late*, Touchstone imprint of Simon & Schuster Inc., New York, paperback edition, 1998.

62. F.E. Heart, R.E. Kahn, S.M. Ornstein, W.R. Crowther, and D.C. Walden, "The Interface Message Processor for the ARPA Computer Network," in *AFIPS Conference Proceedings*, vol. 36, pp. 551–567, AFIPS Press, June 1970.

63. Severo M. Ornstein, *Computing in the Middle Ages*, 1stBooks, network address `www.1stbooks.com`, 2002.

64. Severo Ornstein, e-mails of April 20, 2010

65. S.M. Ornstein, F.E. Heart, W.R. Crowther, H.K. Rising, S.B. Russell, and A. Michel, "The Terminal IMP for the ARPA Computer Network," in *AFIPS Conference Proceedings*, vol. 40, pp. 243–254, Montvale, NJ, AFIPS Press, May 1972.

66. Sources of information for this section were Dave Fosdick (e-mail of July 10, 2003), Phil Herman (e-mail of July 9, 2003); Bob Howard-Anderson (e-mail of July 15, 2003); Mike Kraley (e-mail of July 9, 2003, who also provided a detailed time line of activities from 1975 to 1987); and Randy Rettberg (e-mails of February 28, 2003, and February 23 and 24, 2004). Although slightly paraphrased and reorganized and not shown in quotes, much of the text of this subsection is based on Rettberg's e-mails.

67. M.F. Kraley, R.D. Rettberg, P. Herman, R.D. Bressler, and A. Lake, "Design of a User-Microprogrammable Building Block," in *Proceedings of the 13th Annual Microprogramming Workshop*, IEEE, New York, 1980.

68. R. Weissler, M. Kraley, and P. Herman, MBB Microprogrammer's Handbook, BBN Report 4268, February 1, 1980.

69. The MBB used new dynamic RAM ICs with which earlier users had experienced trouble with occasional errors (later diagnosed as soft errors due to radiation in the package). The MBB designers implemented error detection and correction to overcome these problems, and implemented remote reporting across the ARPANET so they could see if enough errors were occurring to justify the cost of the extra hardware.

70. R. Greene, M. Kraley, and P. Herman, Design Plan for an MBB-Based ARPANET-Like IMP, BBN Report 4090, March 1, 1979; R. Weissler and P. Herman, MBB IMP System Test and Verification Plan, BBN Report 4178, Auguest 15, 1979; P. Herman, R. Weissler, and S. Geyer, MBB-IMP Operator's Reference Guide, BBN Report 4267, November 1, 1979.

71. See also Chapter 6, page 105.

72. Sources of information for this subsection were Jim Calvin (e-mails of March 6 and July 8, 2003); Harry Forsdick (e-mails of March 7 and July 10, 2003); Alice Hartley (e-mail of July 9, 2003); and Ray Tomlinson (e-mails of March 3 and July 10, 2003).

73. Internal Research and Development funding; that is, paid for by BBN itself.

74. See section 21.5.

75. Madeleine Bates and Robert J. Bobrow, "Information Retrieval Using a Transportable Natural Language Interface," in Jennifer J. Kuehn, ed., *Research and Development in Information Retrieval, Sixth Annual International ACM SIGIR Conference*, National Library of Medicine, Bethesda, Maryland, June 6–8, 1983, pages 81–86. ACM, 1983.

76. H. C. Forsdick, Jericho-Pascal Virtual Machine Design, BBN Report 4241, January 1, 1980.

77. Cronus and Diamond are described in section 18.4.

78. Information for this subsection came from Bob Brown (e-mails of July 15 and 30, 2003); Bernie Cosell (e-mails of July 9 and 10, 2003); Rich Fortier (e-mail of July 15, 2003); Mike Kraley (e-mails of March 1 and June 12, 2003, who also provided a time line from 1975 to 1987); Randy Rettberg (e-mail of March 27, 2003); and Robert Wells (e-mail of June 12, 2003, based on his detailed contemporaneous lab notebooks, and e-mail of July 13, 2003). In some cases I have paraphrased and reorganized text from Wells's extensive e-mails without showing it in quotes.

79. Kraley says he also did some hardware work, and Rettberg remembers Carvey's "discussions of 'magic numbers' horizontal frequencies. . . ."

80. BBN Computer Corporation, Cambridge, MA, *BitGraph Advanced Graphics Terminal User's Guide, Version 3.2.* Uncertain original publication date; the first, second, and third revisions were dated March and November 1982 and June 1983. There was also a BitGraph Programmer's Manual (BBN Computer Corporation document 032T002).

81. Pac Man was developed by Bob Brown, then a grad student at Purdue. Before Pac Man, he adapted Alien (which, Brown says, "was floating around the UNIX community at the time") to BitGraph. He also wrote some additional BitGraph development software to support his hacks. He sent Pac Man to Fortier or Wells, and Pac Man swept BBN, as it apparently did other sites with BitGraphs.

82. The BitGraph developers put a video game sound chip in the machine to provide the "beep" sound that all terminals have. In one "neat" hack, Bernie Cosell pulled together some software to produce multipart music on the BitGraph. Cosell started with an early version of Bob Brown's music compiler that input a music description in ASCII that allowed for chords, multiple voices, and dynamics and drove the BitGraph sound generation capability; Cosell also added features from Peter Sampson's famous PDP-1 music program[4] which Cosell had adapted to BBN's PDP-1d years before. He did fairly fancy transcriptions of a bunch of Christmas carols and had the BitGraph set up so that it played them as a "performance."

83. `comp.terminal.bitgraph`

84. David Mankins and Daniel Franklin, "A Simple Window-Management Facility for the UNIX Time-Sharing System," in *USENIX Conference Proceedings*, pp. 203–228, summer 1983.

85. Who, after Purdue, worked at NASA Ames, where they bought several BitGraphs.

86. Phone conversation of September 26, 2002.

87. The first BBN annual report in 1966 also says the Data Equipment Division also built Teleputer consoles and controllers for time-shared computer systems and Dynacontrol control and actuator systems for high-performance hydraulic servomechanisms. In 1971 Data Equipment was sold to a company in Wilmington, Massachesetts, and Keast moved to Massachusetts to be the chief engineer for the purchasing company. In 1973 Keast left that company and was rehired into BBN's acoustics activities.

88. E-mail of March 1, 2003.

89. E-mail of January 25, 2004.

90. Bob Clements was also involved with development of a related terminal concentrator system — an early and very robust implementation of TCP/IP and an excellent terminal handler — that we will not describe here.

91. Lawrence D. Sher, "The Oscillating-Mirror Technique for Realizing True 3D," Chapter 11 in *Stereo Computer Graphics and Other True 3D Technologies*, David F. McAllister, ed., Princeton University Press, 1993.

92. Paul Wexelblat recounted in an e-mail of April 26, 2004, that years earlier Wally Feurzeig sent Frank Fraser and Paul out to buy embroidery hoops to play with the reflective film that was being put on BBNs windows so they could experiment with a 3-D display technique they had seen proposed in an article. Later, Paul reports, he wrote for Larry Sher the first actual SpaceGraph code, two lines in Fortran.

93. The information in this section came from many people: Tom Blackadar (e-mails of May 10–11, 2010); Steve Blumenthal (who in turn consulted with Bob Hinden, e-mail of May 24, 2004); Guy Fedorkow (e-mails of May 11 and 24–25, 2010); John Goodhue (who in turn consulted with Guy Fedorkow and Tim Donahue, e-mails of March 14, 2003, May 22–23 and July 12, 2004, and May 24–25, 2010); Mike Kraley (e-mails of March 1 and July 10, 2003, and May 24, 2010); Bill Mann (e-mail whose date is lost); Severo Ornstein's 2002 memoir[63] (pp. 190–192 and 204–209); Randy Rettberg (e-mails of February 13 and 28, 2003, and February 23–24, 2004); and John Robinson (e-mails of March 1, 2003, and May 25, 2010). John Goodhue provided a particularly thorough review and revision of the subsections "Commercialization of the Butterfly" and "A second attempt."

94. As Severo Ornstein suggests on page 190 of his memoir.[63]

95. For a long time, it was an article of faith among BBN's parallel processing people that there was (or at least should be) an insatiable appetite for computer power in the world at large.

96. A 21-page (probably incomplete) bibliography exists of papers and other documents describing BBN's parallel processing work: `www.walden-family.com/waterside/bbn/pp-biblio.pdf`

97. Many if not most of the series of multiprocessors were used in networking applications. This chapter focuses on the the parallel processing side of things and leaves the networking applications to Chapter 17.

98. F. E. Heart, S. M. Ornstein, W. R. Crowther, and W. B. Barker, "A New Minicomputer/Multiprocessor for the ARPA Network," in *AFIPS Conference Proceedings*, vol. 42, pp. 529–537, AFIPS Press, June 1973.

99. On pages 190 and 207 of his memoir,[63] Ornstein recalls that the multidimensional punny name Pluribus won over ACM (the Association of Computing Machines).

100. His memoir[63] (p. 191).

101. Ornstein also tells in his memoir[63] (pp. 204–209) some of the story of learning about the SUE from Lockheed, the nonstop reliability goals for the new packet switch, and so on.

102. Some may wonder why the team did not adopt the more "obvious" (for the time) choice of the PDP-11. According to John Goodhue, some participants remember there being a "unibus deficiency" with the PDP-11, and some participants remember DEC's not being willing to share enough design information.

103. See Chapter 4.

104. As so often happened over the years, lots of the cleverness of the software came from Will Crowther.

105. D. Katsuki, E. S. Elsam, W. F. Mann, E. S. Roberts, J. G. Robinson, F.S Skowronski, and E. W. Wolf, "Pluribus — an Operational Fault-Tolerant Multiprocessor," *Proceedings of the IEEE*, 66(10):1146–1159, 1978.

106. S. M. Ornstein, W. R. Crowther, M. F. Kraley, R. D. Kraley, A. Michel, and F. E. Heart, "Pluribus — a Reliable Multiprocessor," in *AFIPS Conference Proceedings*, vol. 44, pp. 551–559, AFIPS Press, May 1975.

107. The ARPANET packet switch.

108. An ARPANET IMP that also provided a terminal concentration function.

109. W. F. Mann, S. M. Ornstein, and M. F. Kraley, "A Network-Oriented Multiprocessor Front-End Handling Many Hosts and Hundreds of Terminals," in *AFIPS Conference Proceedings*, vol. 45, pp. 533–540. AFIPS Press, June 1976.

110. When Nancy Mimno joined the TIP team, she called the HSMIMP system HRIMP.

111. The CCP was used in a system that did transmission and analysis of seismic data.

112. For which Jane Barnett was the first programmer.

113. US4130865, Crowther, Heart, Barker, and Ornstein, `http://www.patents.ibm.com/details?pn=US04130865`

114. BBN Reports 2999, 2930, 3001, 2931, 3002, and 3004.

115. E-mail of March 1, 2003.

116. Randall Rettberg and Robert Thomas, "Contention Is No Obstacle to Shared-Memory Multiprocessing," *Communications of the ACM*, 29(12):1202–1212, December 1986.

117. The connection network itself was not new. Rettberg explains that "interconnection networks (e.g., Banyan, etc.) were invented before we worked on them. We also were overly proud of the Butterfly switch. In fact we always drew it backwards. If you draw it reversed, you see that it is a series of switches that always selects which quarter of the nodes contains the target. Then you more easily realize that this is simply a cascade of crossbar switches and the number of inputs and outputs does not matter and the wiring should be random rather than well structured. We did all of this when we designed the Monarch switch" (see page 544).

118. In September 1981 John Goodhue took over from John Pershing the job of developing the Voice Funnel software. Goodhue says, "The Voice Funnel was a gateway to the Wideband Packet Satellite Network for the experimental IP telephones and video conferencing stations being developed by DARPA packet voice community. In addition to IP, the Voice Funnel supported ST — an early protocol for packet voice and video that supported call setup, resource reservation, conferencing, and other facilities that are now part of the standard Voice-over-IP stack. The Voice Funnel was the first application to be deployed on the Butterfly. The first call through a Voice Funnel was made in 1982 between Lincoln Lab and ISI."

119. Metal Oxide Semiconductor Implementation Service.

120. Very Large Scale Integration.

121. See Chapter 18.

122. Will Crowther and Bob Thomas, The Uniform System Approach to Programming the Butterfly Parallel Processor, BBN Report 6149, March 6, 1986.

123. See page 544.

124. See Chapter 17.

125. W. Edmond, S. Blumenthal, A. Echenique, S. Storch, and T. Calderwood, "The Butterfly Satellite IMP for the Wideband Packet Satellite Network," *Proceedings of the ACM SIGCOMM Conference on Communications Architectures & Protocols*, 1986, pp. 194–203.

126. See page 427.

127. W. Crowther, J. Goodhue, E. Starr, R. Thomas, W. Milliken, and T. Blackadar, "Performance Measurements on a 128-Node Butterfly Parallel Processor," *Proceedings of the 1985 International Conference on Parallel Processing*, 20–23 August 1985, pp. 531–540.

128. See Chapter 6

129. Randall Rettberg, William Crowther, Philip Carvey, and Raymond Tomlinson, "The Monarch Parallel Processor Hardware Design," IEEE *Computer*, April 1990, cover and pp. 18–30.

130. In conjunction with the hardware effort, DARPA funded BBN to develop a parallel LISP, to be prototyped on the Butterfly but ultimately targeted to the Monarch; D. Allen, S. Steinberg, and L. Stabile, "Recent Developments in Butterfly LISP," *AAAI-87 Proceedings*, `http://www.aaai.org/Papers/AAAI/1987/AAAI87-001.pdf`

131. E-mail of April 26, 2010.

132. C. Partridge et al., "A Fifty Gigabit Per Second IP Router," *IEEE Journal of Transactions on Networking*, vol. 6, no. 3, June 1998, pp. 237–248.

Chapter 22

Epilog

David Walden

Most of the chapters of this book were drafted in approximately 2003 for special issues of the IEEE Annals of the History of Computing *on computing at BBN. Furthermore, the BBN history in those special issues ended in about 1990 because of the journal's guideline that history is at least 15 years old. For this book any content from 1990 to 2003 that was dropped for the* IEEE Annals *has been added back. This epilog sketches the ongoing evolution of BBN in the 2000s.*

The first section of this epilog continues from the point of Steve Levy's section 6.6 discussion of BBN in the 1990s. That section, which starts on page 116, sketches the BBN transitions from the arrival of George Conrades as BBN CEO until the 1997 sale of BBN to GTE.[1] The second and third sections of this epilog provide a more general picture of how BBN changed between 1997 and 2010.[2]

22.1 Changes in ownership

BBN under George Conrades as president and CEO invested heavily in expanding the market share of its Internet business, BBN Planet. This was in the era of the dot com boom. In time the need for continued investment in BBN Planet resulted in the sale of BBN to the telephone company GTE which wanted to be in the Internet business. In the two years following GTE's June 1997 acquisition of BBN, GTE continued to operate BBN Systems and Technologies (the contract R&D business) and BBN Planet as separate businesses, investing over $1 billion in growing BBN Planet.

In the spring of 1999, as part of the continuing consolidation and evolution of the communications industry, GTE and Bell Atlantic announced that they had agreed to merge their two companies (in other words, Bell Atlantic acquired GTE). However, Bell Atlantic was one of the Regional Bell Operating Companies (RBOCs) that resulted from the historic breakup of AT&T Corporation and was forbidden under the terms of the breakup from being in the "long-distance service" business. BBN Planet's Internet business was deemed to be a long distance business by the Federal Communications Commission; consequently, prior to effecting the merger with GTE, Verizon had to relinquish control of BBN Planet.[3] Verizon accomplished this divestiture of control by allowing BBN Planet to sell $2 billion of its shares in the public market. The resulting public company was named Genuity.[4]

At the time of Genuity's IPO in the summer of 2000, BBN Technologies[5] continued to operate under its own name as a wholly owned business unit of Verizon Communications. Then in February 2004, BBN Technologies became a privately held company again, having been acquired from Verizon by private investors (primarily Accel Partners of Palo Alto, California, and General Catalyst Partners of Cambridge, Massachusetts) and the management of the company. In October 2009, the 2004 investors cashed out of their investment, selling BBN to Raytheon for $350 million.[6]

22.2 The classic BBN culture and business

Not surprisingly, when George Conrades came to BBN as CEO, he brought in new senior managers (see Figure 6.11) with skills he didn't see in traditional BBN managers, many of whom had come up through the technical side of the company. In particular, the position of Frank Heart[7] had as president of BBN Technologies was taken over by David Campbell who came from an executive position outside of BBN.[8] Campbell brought in additional key managers from outside BBN, reorganized BBN Technologies, and in general worked on redirecting BBN Technologies's business in ways he thought appropriate.

When BBN was acquired by GTE in June 1997, David Campbell became a senior executive of the GTE Technology Office, managing GTE Laboratories and a few other GTE held activities, and still personally managed BBN Technologies. Campbell did the job he was assigned and tried to fit BBN Technologies into the GTE culture and organization.

As a result of such reshaping — and the prior emphasis within BBN from 1995 to 1997 on exploiting BBN's Internet activities and changing BBN Technologies' business direction — BBN Technologies' traditional research and development business suffered. Many good researchers and managers left, and for several years the company did an insufficient job of recruiting the potential stars of tomorrow.[9]

Nonetheless, a number of influential BBN Technologies old timers (and an influential consultant) were committed to preserving the "classic" BBN culture and business. They were able to convince David Campbell that BBN Technologies needed its own dedicated leader. In the first half of 1998 a committee consisting of one long-time, senior BBN person and two outside people who knew BBN Technologies' traditional strengths undertook a search for a dedicated leader of BBN Technologies.

Ed Starr, who had joined BBN in 1959 and was serving as part of Campbell's top management team, was chosen to be president of BBN Technologies. Starr was well known and respected throughout the company, having worked as a project leader and business leader in many capacities all around the company. However, Starr was planning to go to half-time work the next year and was thinking about full retirement. Starr agreed to serve as president for 18 months; and Tad Elmer was designated as Starr's successor.

Elmer also had been on the search committee's list, but he had never run a company-wide activity. Elmer was a department manager who had been with BBN for many years and also was well known and respected throughout the company. Elmer had demonstrated entrepreneurial capability, having initiated a new branch office and moved his department into new business areas, particularly at the intersection of BBN's involvement with computers and acoustics. Elmer worked closely with Starr, watching and learning.

Ed Starr did cut back his hours (and eventually retired), and Tad Elmer became president.[10] With Starr and then Elmer at the helm, BBN Technologies began to reassert its traditional culture and approach to business (and financial viability).

By the time of the Bell Atlantic acquisition of GTE and the spin off of Genuity (what had previously been called BBN Planet), the BBN Technologies business and culture had already been substantially reinvigorated.

Verizon did not try to integrate BBN Technologies into the rest of its business. Instead it treated BBN Technologies benignly, and there was mutual respect between

Verizon and BBN Technologies. BBN did do a little bit of work for Verizon, but generally BBN's technologies were in too far from being off-the-shelf products to be useful to Verizon.

During the almost four year it was part of Verizon, BBN Technologies flourished in its classic business. Visiting BBN during that period, I heard one-time BBN colleagues of mine say that is was like the old BBN again — perhaps better.

In 2003 Verizon needed to change its capital structure in preparation for a big investment in FIOS and began looking for buyers for parts of the company not central to its future, including seeking a buyer for BBN Technologies. Worried about coming under the control of another owner not interested in the classic BBN business, Tad Elmer and his management team were given permission by Verizon to seek investors who would make an equivalent financial offer to keep BBN Technologies what it was; and they found such investors.

The sale, primarily to Accel Partners of Palo Alto, California, and General Catalyst Partners of Cambridge, Massachusetts, happened in March 2004, and was celebrated within BBN Technologies and by retired BBN people who retained a strong emotional attachment to BBN classic business continuing to flourish.

Of course, the outside investors wanted BBN to make money for them. Thus, as had happened so often in BBN's past,[11] there was pressure once again to license technology and to create products in addition to pursuing the traditional contract research and development business. This time BBN Technologies tried to be particularly quick and nimble and to take advantage of the contacts of the venture capitalists who were its major investors.

A new division was created for the licensing and product opportunities led by Alex Laats who came from outside the company with entrepreneurial, licensing, and venture capital experience. Between 2004 and 2009, several product opportunities were pursued. Some examples are:

- PodZinger[12] audio and video search engine
- AVOKE™ system and services to examine telephone interactions from the caller's perspective
- Boomerang sniper detection localization system
- Digital Force Technologies (acquisition of a company with specialized manufacturing capabilities)

Boomerang is an representative example. This system detects incoming small-arms fire and displays the azimuth, range, and elevation of the shooter; it can be installed on a vehicle in an hour. Based on BBN Technologies' prior acoustics and computer technology experience and development work, in 2005 DARPA awarded BBN Technologies a contract to prototype this system. BBN completed a set of prototype systems in 65 days. Eschewing the approach BBN had used sometimes in the past of setting up its own manufacturing activity, BBN Technologies outsourced manufacturing for Boomerang with BBN engineers working closely with the engineers of the a contract manufacturer to modify the design for productization, reliability, and manufacturability. In 2006 the Army ordered over 100 systems, and through 2008 there were additional procurements totaling approximately 10,000 units.

Some of the other projects, for example, Podzinger and Avoke, benefitted from the connections of the ventures capitalist owners of BBN.

Over the same period, BBN Technologies grew and expanded its traditional research and development activities, in combination with the aforementioned product activities.

By the time of the sale of BBN Technologies to Raytheon in late 2009, BBN Technologies had doubled its yearly revenue, tripled its yearly profit, and paid off the debt resulting from the leveraged buyout from Verizon.

22.3 The evolution continues

The day before my April 1, 2010, interview with Tad Elmer and Steve Milligan, I studied the BBN Technologies website, `bbn.com`, to try to understand how the areas in which BBN Technologies does research and development had changed since most of the chapters in this book were drafted in 2003. Looking under "technologies" on the website, I found the following categories:

- Advance Networking
- Cyber Security
- Heathcare Informatics
- Immersive Learning Technologies
- Information and Knowledge Technologies
- Sensor Systems
- Speech and Language Technologies

Each of those areas listed between 5 and 30 subareas. While I could see some overlap with what I knew from 2003, much was different.

Tad Elmer explained that he believes in rearranging existing technical groups and adding new groups with fair regularity — to pursue new technology opportunities, especially at the intersections of previous technology areas where innovation so often happens.[13] Furthermore, long-time areas of BBN expertise have expanded. For example, Elmer explained that the sensors and detection area (see Chapter 10) has expanded to detection using any sensor media (for example, sound, infrared, magnetism) to look into or through any sort of substance; and speech and language technology leaders John Makhoul and Ralph Weischedel already hinted (see Chapters 14 and 15) at expansion in their area. There also has been expansion in other areas.

I asked Elmer and Milligan if there were any principles that guide the ongoing evolution of BBN Technologies and development of its people. Milligan said that the senior technical leaders (the "principle investigators" who represent the company technically to customers and mentor the relevant technical people) are allowed to move in any direction they want as long as: (a) it is legal and ethical; (b) someone outside the company is willing to pay for the work; and (c) the work is generally fun and interesting for the people at BBN. Elmer said that point b can slightly dominate point c if the contract is big enough.[14]

Such flexibility not only is allowed; it is encouraged. Some senior people who did not enjoy change and moving into new areas and ways of organizing have left the company.

One can't know for certain how BBN Technologies will change as a consequence of its purchase by Raytheon. However, given the company's 62-year history as a preeminent innovator, one can assume that Raytheon and BBN Technologies will work hard to keep this powerful national resource working in its traditional way in an ever evolving set of problem areas.

Notes and References

1. Some of the information in this chapter came from Steve Levy's work on Chapter 6. Steve Blumenthal and Harry Forsdick providing confirming details on the Genuity-to-Level 3 transition in e-mails of April 3, 2010.

2. This information came from an interview of BBN Technologies president Tad Elmer and chief technology officer Steve Milligan at BBN in Cambridge, Massachusetts, on April 1, 2010, and from an e-mail of June 27, 2003 from Ed Starr.

3. GTE's ownership of BBN Planet had not had a "long distance" problem because GTE had not been an RBOC.

4. Verizon retained an option to convert, under certain conditions, the 10 percent ownership it held after the sale of the equity into a 90 percent ownership position in Genuity. Between 2000 and 2002, Genuity continued to invest heavily in its growth, and annual revenues grew to in excess of $1 billion. Nevertheless, after a cumulative investment approaching $6 billion, Genuity was still incurring heavy operating losses; and, in the fall of 2003, Verizon announced that it was not going to exercise its option to convert its 10 percent equity position in Genuity into a 90 percent ownership position. Given its heavy, negative cash flow Genuity was then forced to file for bankruptcy under Chapter 11 of the U.S. bankruptcy laws. In early 2003, Level 3, Inc., acquired the operating assets and business of Genuity for something on the order of $200 million.

5. Somewhere along the line, the "Systems and" part of the BBN Systems and Technologies named was dropped.

6. The 2004 purchase price has never been publicly disclosed, but the 2009 sale price purportedly produced a nice profit for the investors, especially in a time of generally poor economic results.

7. See Chapter 7.

8. Conrades himself led BBN Technologies for over a year, in addition to his many other duties, until he was able to bring Campbell on board.

9. See the discussing of recruiting in section 5.2.

10. Around the same time David Campbell left GTE.

11. See Chapter 6.

12. The system's name was later changed to RAMP.

13. Also, several years earlier BBN's business related to ship quieting had been sold to Raytheon, thus eliminating that part of the company's historic business.

14. In mid 2012 Tad Elmer retired from BBN. According to the `bbn.com` website, Ed Campbell is now president. At the time of his appointment as president, Ed had been with the company for 34 years and had been Tad Elmer's deputy for many years.

The fonts used in the main text of this book are from the extensive Lucida typeface family of fonts designed by Charles Bigelow and Kris Holmes from 1985 onward. The Lucida family was specifically developed with computer output devices in mind.

The sans sarif typeface used in tables and figures is Helvetica which has a long history involving many designers and many additional fonts since its initial font was created in 1957 at the Hass foundry in Münchenstein, Switzerland.

ISBN 978-0-9789737-0-4
90000
9 780978 973704