Abstracting Common Business Rules to Petri Nets (Abstract)

Kees van Hee[1], Jan Hidders[2], Geert-Jan Houben[2], Jan Paredaens[3], and Philippe Thiran[4]

[1] Eindhoven University of Technology, The Netherlands
[2] Delft University of Technology, The Netherlands
[3] University of Antwerp, Belgium
[4] University of Namur, Belgium

These days Business Information Systems (BIS) have a highly responsible task: they execute large parts of the *business processes* autonomously. So organizations become more and more dependent on their BIS. Business processes usually require a certain order in which activities have to be executed. On top of that, there are many other *business rules* that should be met by the execution of business processes, e.g., the famous Sarbanes-Oxley rules (cf. [1]) and the General Accepted Accounting Principles (GAAP). Business rules can be required by any stakeholders, such as management, government, shareholders and business partners, both clients and suppliers. In this paper we take it for granted that a BIS has some unknown errors and that the system is for all stakeholders more or less a black box. Instead of a human auditor we propose here an approach where the BIS is extended by an independent *monitor system* that checks the essential business rules on the fly and that reports violations of business rules or that interrupts the BIS to prevent the occurrence of a violation.

As basic terminology we use workflow notions such as *task*, *case*, *agent* and *resource*. This is consistent with the terminology for *accounting information systems* where the REA (Resource, Event and Agent) data-model is used frequently. This model is an Entity-Relationship model for accounting information systems [2–4]. We assume that business processes are described by traces which are here defined as sets of events where events are partial functions that map event properties to property values. Event properties can be either a standard mandatory property such as **time**, **case**, **task**, **agent** or a certain type of resource that is involved in the event such as money or used materials. A property value can be either a task name, an agent, a basic value, or a rational number where the latter is also used to represent time stamps. The values of standard properties are restricted so they have property values of the right type.

The syntax of the BRL, the business rule language we present here, is defined by the following abstract grammar:

$$F ::= \neg F \mid (F \wedge F) \mid \langle P_S = E, \ldots, P_S = E \rangle \mid E = E \mid E < E.$$
$$E ::= \mathcal{X} \mid D_S \mid \Sigma(\mathcal{X} : F)E \mid (E + E).$$

Here F denotes boolean formulas and E expressions that return a property value. The non-terminals P_S and D_S represent event properties and property values, respectively, and $\mathcal{X}$ represents variables that range over property values, which includes time-stamps. The value of the formulas and expressions are defined w.r.t. to a certain trace. There are three special workflow constructs with specific workflow semantics: formulas of the form $e_1 < e_2$, formulas of the form $\langle p_1 = e_1, \ldots, p_k = e_k \rangle$ and expressions of the form $\Sigma(x : \varphi)e$. The first compares the values of e_1 and e_2 according to the ordering defined over the property values, which includes the time ordering over time-stamps. The formula $\langle p_1 = e_1, \ldots, p_k = e_k \rangle$ states that the trace contains at least one event ev such that $ev(p_1) = v_1, \ldots, ev(p_k) = v_k$ where $v_1, \ldots, v_k$ are the values of the expressions $e_1, \ldots, e_k$ for the trace. The expression $\Sigma(x : \varphi)e$ is a sum quantifier which sums the values of expression e for each property value x that occurs in the trace and satisfies the formula φ. For example, $\Sigma(x_t : (x_t = x_t))(\Sigma(x_p : \langle \mathbf{time} = x_t, \mathbf{payed} = x_p \rangle)x_p)$ computes the total amount of money payed.

The sum quantifier allows us to formulate business rules about aggregates of used and produces resources such as "for each case at each moment the total amount of used bread does not exceed the amount of previously delivered bread". In addition it can also be used to formulate more standard workflow rules concerning precedence, such as "an event for task a is always followed by an event for task b". For example, the formula $(\Sigma(x : \langle \mathbf{time} = x, \mathbf{task} = a \rangle)1) > 0$ checks if at least one event for task a has occurred.

In the full paper it is shown how for certain types of rules in BRL we can define a monitor system that is able to check them on the fly and in parallel to a business information system. This is done by translating parts of the evaluation of business rules into the execution of a Petri net [5] by using *history-dependent Petri nets* [6]. This is an efficient way of checking rules because it can be done in an incremental way, i.e., event by event, using a Petri net engine. If the Petri net cannot execute a transition, then a rule is violated and this can be reported, or it may generate an interrupt for the BIS and trigger an exception handler. In the future we will also try to transform larger classes of business rules.

References

1. Berg, D.: Turning Sarbanes-Oxley projects into strategic business processes. Sarbanes-Oxley Compliance Journal (2004)
2. Romney, M.B., Steinbart, P.J.: Accounting Information Systems. 11 edn. Pearson International Editions (2009)
3. Geerts, G.L., McCarthy, W.E.: An ontological analysis of the economic primitives of the extended-REA enterprise information architecture. International Journal of Accounting Information Systems **3**(1) (2002) 1–16
4. McCarthy, W.E.: The REA accounting model: A generalized framework for accounting systems in a shared data environment. The Accounting Review **57**(3) (1982) 554–578
5. Reisig, W.: Petri Nets: An Introduction. Volume 4 of Monographs in Theoretical Computer Science: An EATCS Series. Springer-Verlag, Berlin (1985)
6. van Hee, K., Serebrenik, A., Sidorova, N.: Token History Petri Nets. Fundam. Inf. **85**(1-4) (2008) 219–234