Why Ecma OOXML cannot be regarded as a free/open document standard?

Note submitted to the Working Committee, Board of Indian Standards on WordprocessingXML, a component of OOXML

Nagarjuna G.

16/05/07

nagarjun@gnu.org.in

Table of Contents

1 Introduction

Let me briefly state the objectives of this note. During the working committee meeting on 7th May 2007 at BIS it was decided that I draft the main arguments for, why free software community thinks that OOXML cannot be regarded as a free / open standard. This draft though is based on a number of resources already published, it is an attempt to bring together a few of the most important objections raised against OOXML along with a few other additions that came to my notice while studying the 6000+ page proposal submitted by Ecma[1]. Some of the details already published (for example in footnote 5.) are not included in this document.

All the comments expressed below are based on what the author's understanding and assumptions of what is a free document standard. I will therefore provide such a statement in section 3 before listing the arguments. The rationale for using the term 'free' in place of 'open' is also to bring to the notice of the standards body the opinion of Free Software Foundation. Briefly stated, the rationale is: *opening is not enough, without the freedom to implement.*

Let me begin by appreciating Microsoft for taking the responsibility by providing a data exchange format in which several million documents could now be converted into an encoding that is not apparently private. This action can save a lot of public as well as private documents from vanishing. But, before a non-private encoding becomes open and finally free, it is a long way. Though Ecma and MS pushed this non-private format to ISO to get a standard status, the existing situation does not seem to be in their favor.

It is however very important to keep in mind that ISO is a standards body and not an exclusive 'open standards' body. A standard granted by ISO does not imply that it is free or open. Therefore satisfying the ISO conditions is not enough for free software community.

2 What is a Free/Open Document Standard?

I present here the following requirements, which according to me make a proposal a free document standard (hence forth FDS). In order to qualify as an FDS a proposal must meet all the following applicable conditions. In the following section I will try to demonstrate that OOXML does not pass through this filter, while ODF does[2]. The criteria, obviously, are not invented for excluding OOXML but based on sound reasoning keeping in mind the requirements of a good digital society.

1 Decodability

One of the **first condition** to be met by a FDS is that the data is encoded by any application according to an explicitly stated and published convention such that any document saved in such a format can be decoded (interpreted) as per the published specification. This

[1] Ecma 376, 1st Edition December 2006

[2] I am deliberately using the term 'Free' in place of 'Open', since a few of the open standards pass all but not the condition stated in 3.7.

condition ensures that the data created by the users in an electronic format is accessible (interpretable) not only in the application in which it is created, but in all future versions of the same application, as well as in all other applications that implement the standard. This ensures not only the endurance of the data, but also its use (interpretation) in future. Since an official document created by any agency for egovernance (Government or any public/private body) is legally required to have such endurance, any proposal must meet this condition.

The most widely used term for this condition is 'inter-operability'. The term 'decodability' is used in place of 'inter-operability' since the latter is not possible without the former. It is scientific to use an operational term, for operational term does not have multiple-interpretations.

2 Use of preexisting standards

The proposed standard must use existing standards of the similar kind. In the context of a document standard it is important to remind ourselves that a document is a collection and composition of *codes*. All code is arbitrary to begin with, till it becomes a socially accepted convention. Standardization is the process by which we arrive at such social acceptance. It is possible to invent several distinct arbitrary codes to represent the same thing. This latter possibility is good for expressing creativity, but is not desirable to indulge in such an activity if reusability and endurance of electronic documents is our primary practical objective. If reencoding is required for technical or practical reasons, it is important to invent one.

SGML (ISO 8879) is a good example of a standard markup, since such a technical requirement was felt by the industry at that time. When XML was proposed as an open standard, it did not throw away the existing standard, though it was possible to create an arbitrary new language, instead they made use of SGML, included an additional contraint to it, and also adopted another open standard Unicode (ISO 10646). This is how they went ahead by creating an exemplary standard to serve the purpose of multi-lingual document exchange. HTML was re-represented in XML and became XHTML. This example may be considered as a paradigm case of how standards are created and extended to meet technical and practical demands. Let us keep this example while handling the case at hand.

3 No Private Language Components

The proposed standard must not have private language components embedded within it. A private language is an encoding, decoding of which is not declared publicly. Though this sounds like a very important requirement, it is redundant, because by adhering strintly to the first condition we already precluded this possibility. I am committing redundancy in order to make the argument complete and explicit, particularly to handle the case at hand.

4 Implementation Independance

One of the most important objectives of agreeing to some standard is to ensure its implementation by multiple vendors. The objective of the standard should not be defined to conform to a single implementation.

5 Transparent Collaborative Production Process

The standard must be developed by involving and inviting all the stake holders in a transparent collaborative consensus driven process. This is to ensure that the standard is not dominated by any one interest group. There must be a room for inviting contributions, suggestions, criticisms and improvements from any interested agency. Since a standard is required to be upheld by several agencies, it is necessay to take everyone into confidence.[3]

6 Rationale for deviating from the preexisting standards

If there are technical or practical reasons for not using an existing standard, it is necessary to specify the relation or lack of such relation with the existing standards, particularly when the existing standards are already adopted by several agencies. This possibility ensures two things. One it allows inventing new ways of expressing, and two it helps in establishing relations to known canons of understanding. This is the way modern science and technology manages to innovate as well as relate to history.

7 Demonstrate incommensurability

If the innovation is so novel that it is incommensurable with the existing set of conventions, then it is indeed a celebration time, for scientists and engineers indeed look forward to listen to such radical innovations. However, the inventors do have the burden to demonstrate that the new innovation is incommensurable with the existing canons, and why it should be considered for a standard, specifically if their innovation is to be regarded as a standard.

In any case, incommensurability is a very rare case, and such events are exceptional. Exceptional cases are better not handled by standardization process. Because, standardization is about recognizing a procedure, to make a known art into a social convention.

[3] Definition of Open Standards, June 2004, National IT and Telecom Agency, Denmark.

8 Freedom to implement the standard

And last, as usual not the least, is the condition that every agency must be given freedom to implement the specification without royalty. If there exist any restrictions in use due to say some patents or other such rights, they must be explicitly exempted before agreeing to confer the FDS status to a proposal.[4]

There may be many other parameters on the basis of which one may consider if the proposal is a free/open standard. Though I have not made references in formulating them, these are stated keeping the already published ideas and following the several debates on the issue in mind. Given enough time, one can make a comparision to the existing understanding on the subject.

3 Does OOXML meet the above conditions?

Having made the criteria of analysis explicit, let me now see if OOXML meets the above conditions.

1 OOXML is not decodable in all cases

The first condition of decodability is *apparently* met by OOXML. The fact that it is encoded in XML apparently gives one this impression. However, OOXML specification contains certain special provisions to embedd 1. arbitrary custom XML at places, 2. possibility to embedd e.g. OLE objects and 3. possibility to reference to any document encoded in a proprietary format. This enables an an applicatiion vendor to keep proprietary elements within the so called standard.

After the emergence of semantic web standards, like OWL (Web Ontology Language), which is encoded in another open standard called RDF (Resource Description Framework), and the most recent ISO standard CL (common logic) the need for inserting any arbitrary XML in any document is almost nil. These standards are meant to insert any arbitrary annotations to any component of a document. These are based on the wisdom that any description logic (DL) can be captured by first order logic, and therefore, if one vendor encodes in any standard language that can express DL, any other application vendor can make sense of them, even if the assertions made are arbitrary. The wisdom in this is visionary, and is driving the current revolution called semantic web which is transforming the current web publication into Web 2.0.

OOXML does allow the inclusion of any of these standard markups inside the OOXML

[4] This is also an important condition to call a standard 'open' according to Danish Government definition, European union definition and Bruce Perence definition.

document. This is because they gave the freedom to insert any arbitrary XML. This is not wise, and can promote proprietary interests. How this can promote proprietary interests is explained in the following scenario.

Take a vendor who adds a feature which allows the individual users to add annotations in their office documents. The vendor advertises this feature as additional value their customers can get if they purchase their office implementation. Attracted by this nifty feature, users begin to use this feature and add lots of annotations. These annotations are encoded in some arbitrary XML tags in the document, which OOXML allows. The user exchanges this document to another colleague, and to make the matter a little more complicated, to a colleague in another country using another office implementation. The second user cannot make use of the annotations unless the latter also uses the same office implementation. This to me appears a way of inserting the proprietary interests in side an otherwise completely decodable XML.

This is a good example to show how a vendor can *abuse* XML, an open standard, to generate a proprietary encoding. As experts we must understand that this kind of exploitation is possible within XML. Simply declaring that the encoding is XML is not therefore sufficient to call a standard free or open. This is a common misconception that any encoding in XML becomes a standard.

As explained earlier, computer science as well as standard making bodies already siezed of this requirement, and created the standard ways of semantic markup that can be used about a document (as in the case of XTM, XML Topic Maps) or within a document by inserting arbitrary assertions by using RDF or OWL or even more expressive CL.

I therefore argue that no *arbitrary insertions* neither be allowed nor requuired in a FDS. Thus, though on the face of it, OOXML is apparently open, it is cunningly made not to be a FDS. Therefore the first condition is not met.

2 OOXML does not use many of the existing standards

This is the main reason why the OOXML specification runs to over 6000 pages, and ODF which meets exactly the same goal in about 700 pages. The only open standard they have used (rather abused) is XML.

Several published comments on OOXML already make this point sufficiently well, therefore I do not want to repeat them here. Not only that OOXML does not make use of ODF, it also does not make use of MathML, SVG, Xlink, RDF etc.[5]

[5] See also ODF Alliance UK Action Group Technical Distinctions of ODF and OOXML: A Consultation Document by Edward Macnaghten,

All that OOXML demonstrated in this massive effort is the possibility of creating another *arbitrary* markup from scratch. Some claims that they made, such as OOXML is more terse and produces more compact documents, is untenable because XML is meant to be a human readable markup so that developers can implement and reimplement support for a FDS easily. XML exists as an intermediary data exchange format, but at the same time it is also a very easy way of communicating in a desciplined natural language. That is the reason why XML is the industry choice for communicating not only with human beings, but also with machines. This is the second case where OOXML abuses XML.

If terseness is the innovation of OOXML, it is not necessary to create another massive standard. Give me some time, I will create another terse ODF format, and publish a table specifying the one to one correspondence with the existing non-terse ODF. Whether this exercise has any technical or practical value, I doubt. This is not what they did anyway, they unnecessarily created a massive arbitrary code, with the single minded objective to map their own proprietary documents.

As already stated earlier, what they did is good for making their proprietary formats open. It does however serve this purpose, and let me repeat: **this is the only purpose OOXML serves**. Let us congratulate them for this feat. But, this was their responsibility, long due. Having completed their task, they are interesting in pushing this as an open standard. I think civil society should not let this model to become a shortcut process to create a FDS. Please see the latter condition, that the process of creating a FDS must also be a collaborative exercise, as is the case in ODF. It is known that the current vendor boycotted that process, though it was very clear that the vendor was one of the major stake holders of ODF. This instance therefore is a clear case of expression of a big brother attitude, and not a socially admirable act.

3 OOXML allows insertion of private language components

The reason why OOXML is not decodable in every case, is already illustrated above. Here I wish to bring to the notice of the possibility of inserting OLE objects, even embedding and referencing proprietary documents within. As stated already, this is a redundant statement. My only justification is that this point should not escape any one's attention. This defeats the whole purpose of creating a FDS.

They might respond by saying a vendor can add lot of value by embedding objects within a document. Yes, they can. But not without voilating the norms of FDS. This can be achieved by other means,such as in UML. This is technically possible, since UML is based on a standard specification of object orientation OMG.

Embedding and referencing proprietary documents within must be barred from the process, for it defeats the purpose.

4 OOXML is clearly implementation dependent

Ecma's statement, fully quoted below in Section 5, makes this very clear. Their objective is clearly not to arrive at a generic document standard. By providing the need to refer to and embedd non-standard vendor specific components (see the above section 3.3), OOXML makes *full* implemenation by other vendors impossible. If such a specification becomes a standard, it will lead to furthering a monopoly. Since the very objective of a standard specification is to eliminiate this possibility, OOXML can not be regarded a standard.

5 OOXML is not produced collaboratively among the stake holders

The ITU-T definition of open standard stipulates this important condition. This ensures that standardization process is not dominated by one single interest group.

6 OOXML specification does not provide rationale for not using or deviating the existing standards

There is no reference to the existing standards in the specification, and also no comparision. This is the tradition of science and technology, and in a sense the ethos of the modern democratic society. Deviation or extension of the existing standards is possible. For example, CL (common logic) as recent ISO standard, is an extension of KIF (Knowledge Interchange Format) and also borrows from the wisdom of a published tradition of Existential Graphs in the form of CG (Conceptual Graphs). There may be several other examples. This to my understanding is a desirable way of deviating and extending from the already existing standards.

7 OOXML specification does not demonstrate that it is incomensurable with the existing standards

This is clearly not an applicable condition, since OOXML did not invent anything new. According to Ecma's own assertion, ODF and OOXML are overlapping standards, therefore no incommensurability.

8 It is not clear if OOXML implementation is royalty free

In the reponse given by Ecma to the comments (Ecma/TC45/2007/006) it is clear that they are trying to meet ISO's requirements on this issue. What they are doing is what is possible by the standards organizations. This may make it a standard, but not a free standard. The rational to keep standards free is well published. See for example Eben Moglen's appeal to W3C that RAND is not enough for a FDS.[6] European Union's, Danish Government's, Bruce Perence's definition also requires this condition to be satisfied for a standard to become open.

4 Response to EMCA's Response

Ecma published responses after the 30 day review of the fasttrack ballot (Ecma/TC45/2007/006). These responses make one point very clear: Ecma is catering to the special requirements of MS, and the objective is not to produce a standard. Having produced an XML version, they want to push this as a standard.

The existence of alternate standards (ODF, HTML, PDF) for office document, and CGMOpen and SVG for vector graphics, JPG and PNG for rastor graphics, RelaxNG and DTD for schema specification in XML document, and TIFF and PDF for press ready graphics format are quoted as existing precedents that there are multiple standards for the same task. In all these cases, there are some special features one format supports, and the other does not. The users of each of these formats make good decisions as to which format is good for which context/purpose. This itself indicates that if the purposes are different they need to make an appropriate decision.

For example, if one wants to display an image on the screen, any one of JPG or PNG will do (FSF recommends PNG for it is a free standard). But if the user intends to animate only PNG can be used. Thus, in the former usecase there is overlapping, while in the latter usecase only one of them can do the job. This kind of usecases can be shown in all the examples given above.

The issue at hand, the comparison of OOXML and ODF, is not of this kind, for both of them overlap mostly, and their functionality is identical. The few elements that can be found in one and not in the other constitutes the real problem. The effort of the industry should be to identify these elements and find a technical solution.

ODF cannot include arbitrary XML to the best my knowledge. If I am right, then ODF is a purer FDS since by design it does not make undesirable elements to be part of the documents. Without such contraint it is not possible to achieve inter-operability. Ecma says ODF cannot take care of these elements. But why take care of them when this is against the goals of a FDS.

[6] http://lists.w3.org/Archives/Public/www-patentpolicy-comment/2001Sep/0650.html

Therefore all we can see in this analysis is that OOXML and ODF though have a lot of overlapping, they do not overlap in the following features: 1. 100% interoperability which ODF has, while OOXML does not have it by design, and 2. OOXML can embedd private language elements while ODF does not.

Ecma's argument that several of MS's clients would not want to loose important data, while transferring from proprietary format into OOXML, and that is why they want to have other means of embedding private data. This clearly demonstrates that existing proprietary docuements cannot be efficiently and completely converted into inter-operable parts of OOXML. This may be either a failure on their part to produce a fully expressable office document standard, or they do not want to free all elements of their proprietary encoding.

A few things ODF cannot do which OOXML can do. One example is, in the current version of ODF it is not possible to insert a table in a slide. Therefore, if one is making a filter from OOXML to ODF, there will be loss. It is in the interest of the community that we should identify all such problem areas. All such problems can be solved. There is, to the best of my knowledge, nothing inherently wrong in ODF structure that prevents it to solve such problems. The free software community could identify such problems and solve them in the forthcoming releases of the standard. If the features that are missing in ODF are genuine, and are required, Ecma would have proposed their addition in ODF, which I am sure ODF community will welcome.

In Ecma's response document the truth of the matter comes out very vividly:

> OpenXML is designed to represent the existing corpus of documents faithfully, even if that means preserving idiosyncrasies that one might not choose given the luxury of starting from a clean slate. In the ODF design, compatibility with and preservation of existing Office documents were not goals. Each set of goals is valuable; sacrificing either at the expense of the other may not be in the best interest of users. (p.6 Ecma Response)

This is the fact of the matter. This clearly shows that one of them is trying to preserve the existing data created by a single vendor, while the other is to provide a generic encoding standard for office documents. It is true therefore that their purposes are different. Since there is a difference in purpose despite the overlapping with ODF, Ecma argues, OOXML can also exist with ODF.

But the issue is: providing a way of preserving a vendor's old documents is the service that a vendor is expected to do. This must happen. This can happen by converting the documents into ODF. Ecma did not prove that this is impossible.

We therefore think, that Ecma has the burden to prove that proprietary documents made by them cannot be converted into ODF. It is very likely that there can be a few elements that cannot be translated, since ODF was not made to serve a particular vendor's requirements. Once such elements are identified, Ecma can propose a model of extending ODF so that the possible problems are sorted out. This is the desirbale way, so that every office document is available in a version of FDS and free software community's preference is ODF format.

5 Concluding Remarks

1. Based on the analysis, OOXML is not a FDS.
2. It is in the interests of all the users that all the proprietary documents be converted into the existing free document standard, ODF.
3. Ecma or MS may identify the elements that ODF cannot take care, and propose extensions to ODF, or create a standard if necessary that is an extension to ODF, ensuring compatibility.