

Microsoft and Open Standards

Can Other Vendors Implement Microsoft's Office Open XML?

15 August 2007

Authors

Matthew Cruickshank	matthew@holloway.co.nz
Chris Daish	chris.daish@catalyst.net.nz
Conal Tuohy	conal.tuohy@vuw.ac.nz

Acknowledgements

This white paper is - in part - sourced from existing research. Particularly the authors wish to thank Groklaw's offshoot Grokdoc, IBM's Rob Weir, the ODF Alliance, the Open Document Fellowship, and – of course – Microsoft themselves.

Executive Summary

Office Open XML (OOXML) is a specification developed by Microsoft Corporation for future versions of its Office Suite of products (MS Word, Excel, PowerPoint etc). OOXML has been submitted and accepted by Ecma International as Standard ECMA-376¹. At the time of writing, OOXML has been submitted to ISO/IEC to be 'fast tracked' for endorsement as an international standard.

This paper examines whether OOXML can be fully implemented by vendors other than Microsoft and concludes that a number of application specific and undisclosed behaviours (as well as a number of other technical flaws) in the proposed standard make this impossible. Also while Microsoft has waived patent claims for the explicit and required parts of the specification it is clearly stated that this does not extend to the undisclosed behaviours or ambiguous definitions, providing a legal as well as technical barrier to OOXML's implementation.

1 <http://urltea.com/11pr?ooxml-at-ecma>

Introduction

Software Office Suites have traditionally comprised a core set of word processing, spreadsheet, and presentation functions. These functions allow customers to create documents and then serialise them to a file (read: save to a file) or stream them across a network link. The structure of this serialisation has been a closely guarded secret for Microsoft, due to the fear that other vendors who achieve full compatibility with their file format may compete with and ultimately supplant them.

Customers, however, desire competition because of the risks of *vendor lock-in*: dependency on a single vendor, higher switching costs, and the stagnation of product innovation. In the case of Microsoft Office, the term “*upgrade treadmill*” has been coined for customers buying new versions not for their features, but for the continued ability to exchange files with others. In economic terms a *network effect*² in combination with closed formats has compelled customers to buy new software capable of reading the secret serialisation of Microsoft's ubiquitous defacto-standard. Due to an increase in understanding about the scope of file formats (and how they affect governments, businesses, and millions of New Zealand home users), there has been a ground swell of support for open standards. Customers are increasingly relying on open standards (as opposed to Microsoft's defacto standards) to provide them choices and cost savings³.

We are therefore happy to see that Microsoft has chosen to listen to its customers and propose an open standard. This white paper covers the subject of whether Microsoft's proposed open standard can help customers as other open standards have in the past. In other words, *can other vendors implement Microsoft's Office Open XML?*

This white paper analyses whether vendors can possibly implement areas of the OOXML specification relating to:

1. **Word Processing Formats**
2. **Spreadsheet Formats**
3. **Presentation Formats**

Please Note This white paper covers *some* of the issues vendors will face when implementing OOXML, but certainly not all. Readers evaluating OOXML should also consult the resources listed in *Appendix A: Other OOXML Analysis*

² <http://urltea.com/11c8?wikipedia-network-effect>

³ Although New Zealand revenue for the Microsoft Office Suite is not known the internationally revenue from the suite is approximately 3.8 billion per quarter. See <http://urltea.com/11k6?microsoft-financials>

1. Word Processing Formats

1.1. Historical Compatibility

OOXML contains compatibility markers to describe older legacy documents, their quirks and processing models. These compatibility features mark behaviours that software must implement to correctly display and process the majority of documents in existence.

The "Compatibility Settings" WordProcessingML4 section within OOXML does not provide for repeatable practices. While it provides Microsoft the ability to store information related to various behaviors in their legacy file formats, the specification merely lists the names of these settings without proper definitions. An OOXML-consuming application, presented with a document using these attributes, will be unable to interpret them properly and render the page in a high-fidelity manner. Further, since these attributes are merely listed but not defined, the ability to practice the benefit of being "fully compatible with the large existing investments in Microsoft Office documents" (the goal of OOXML according to its authors) is consequently reserved for Microsoft alone.

<http://urltea.com/v9w?> [240KB, PDF]

These behaviours such as "*autoSpaceLikeWord95*", "*useWord97LineBreakRules*" and "*useWord2002TableStyleRules*" are not defined. As OOXML repeatedly states,

[t]o faithfully replicate this behavior, applications must imitate the behavior of that application, which involves many possible behaviors and cannot be faithfully placed into narrative for this Office Open XML Standard.

OOXML, ECMA-376 Part 4, Page 1384, 2.15.3.6

These processing hints in the proposed standard depend on undisclosed information, and therefore other vendors cannot correctly process historical documents using OOXML.

This lack of specification has significant implications for the New Zealand public sector organisations operating under the Public Records Act who are seeking to preserve documents of their records in a readable electronic form.

1.2. Graphics

1.2.1. Vector Graphics and Microsoft Patents

Use of vector graphics improves the clarity of both print and on-screen display by providing a resolution independent way of defining images. OOXML defines vector graphics using two formats: DrawingML and VML (Vector Markup Language). One such example is the DrawingML format's

<shape> element.

Section 6.1.2.19 p. 5438 defines a "gfxdata" attribute for the "shape" elements, which "contains DrawingML content" that is "base-64 encoded". However, the "contents of this package are application-defined", so even though they "shall use the Parts defined by this Standard whenever possible" there is not sufficient information for an independent implementation to read this data or display the "DrawingML content" contained therein.

<http://urltea.com/xo0> [140KB, HTML]

Since the proposed DrawingML standard depends on undisclosed information, other vendors cannot reliably use documents with vector images.

The other vector graphics format described in OOXML, VML, appears to be covered by New Zealand patent #525857, "Markup Language And Object Model For Vector Graphics". Several wider legal analyses of OOXML's patent grants⁴ have found that they guarantee no explicit rights. What is granted is use of

Microsoft-owned or Microsoft-controlled patents that are necessary to implement only the required portions of the Covered Specification that are described in detail and not merely referenced in such Specification.

<http://urltea.com/11gg?microsoft-open-spec-promise> [40KB, HTML]

However as many parts of the specification are deemed non-required by Microsoft (optional components) it's clearly stated that vendors other than Microsoft can only implement a subset of OOXML without infringing patents. That required parts of the proposed standard are undisclosed and therefore not "described in detail" not only technically prevents other vendors, but it also legally encumbers any vendor wishing to implement OOXML. As a consequence anyone attempting to fully implement the OOXML specification lays themselves and their customers open potentially to legal action by Microsoft.

Further, there are some concerns raised in a later statement in the patent grant:

[n]o other rights except those expressly stated in this promise [respectively, covenant] shall be deemed granted, waived or received by implication, or estoppel, or otherwise.

<http://urltea.com/11gg?microsoft-open-spec-promise> [40KB, HTML]

This grammatical construct was criticised due to it's ambiguity and yet it remained unchanged in the final OOXML specification. (For more information on this please refer to <http://urltea.com/11bx> .)

Alternate serialisations of OOXML may infringe on Microsoft's New Zealand patent #525484 "Word-processing document stored in a single XML file that may be manipulated by applications that

⁴ Microsoft names their patent grants an "Open Specification Promise" in a "Covenant Not To Sue". See <http://urltea.com/11gg>

understand XML".

1.2.2. Bitmap Graphics

Bitmap graphics (also known as "raster images") are typically rectangular grids of colours used to create images at a fixed resolution.

Section 6.2.2.14 on p. 5596 defines an "ink" element which stores "ink annotations in an application-defined format." This is apparently intended to store Microsoft Ink annotations, used with tablet input devices to add hand-written annotations to documents. These annotations are often a vital part of documents and their specification is undefined in Ecma 376. Microsoft [...] reports that it uses one of two proprietary formats for Ink content: an Ink Serialized Format (ISF) encoding the user's pen-stroke information as well as other metadata (using an undocumented compressed format), as well as a "fortified" GIF format including ISF meta-data.

<http://urltea.com/ox0> [140KB, HTML]

These two features of OOXML affect graphical display, printing, and processing models. Because these features rely on undisclosed formats vendors cannot reliably or accurately implement documents that use images in this way.

1.3. Lists

OOXML covers the styles of many bulleted and numbered lists, however these styles are merely given names rather than a specification. For example, OOXML specifies "Korean Legal Numbering" without stating what it means or how to practise it. Elsewhere a list style of "chicago" is defined as

"[a] sequence [that] shall consist of characters as defined in the Chicago Manual of Style".

There have been 15 editions of the Chicago Manual of Style, and no edition or page is reference provided in OOXML. Styles such as "ideographDigital", "koreanDigital2" and "heebieJeebies" suffer the same lack of detail. Although the details of implementation are known to Microsoft, in their source code, these details have not been revealed in OOXML.

Because this information has not been disclosed other vendors would encounter difficulties implementing lists within the proposed OOXML standard.

1.4. Security

OOXML uses cryptographic functions for passwords and digital rights management (DRM). In America the Federal Information Processing Standards⁵ (FIPS) and National Institute of Standards and Technology (NIST) define a list of approved cryptographic functions. ISO too have ratified

⁵ <http://www.itl.nist.gov/fipspubs/>

functions such as SHA-256 in ISO/IEC 10118-3:2004.

OOXML defines several cryptographic algorithms which are not among those approved for use by NIST in their FIPS-180 list of compliant algorithms. Instead of using a recommended algorithm like SHA-256, Microsoft specifies a legacy hashing algorithm, presumably used in earlier versions of Microsoft Office.

They provide DRM-based protections in Office 2007. These are not documented in OOXML, so no other vendor is able to freely use those features. Instead we're left with the flawed legacy security support of OOXML, support which is not even FIPS-180 compliant.

<http://uritea.com/v9w?technical-case-against-ooxml>

As these DRM features are undefined in OOXML other vendors cannot interoperate with DRM documents created by Microsoft Office.

Although this bears little relation to a technical analysis of OOXML it's of note that Microsoft's OOXML specification does not standardise a subset of existing and future Microsoft Office documents. Competitors can, therefore, not interoperate with these documents due to a lack of specification.

1.5. *Default Behaviour*

Numerous elements are not required by the proposed standard. If omitted, however, these elements lead to "application-defined" default behaviors. These undefined default behaviours will cause the same document with "default" styles to appear differently in two conforming programs.

This issue could be avoided in future versions of OOXML by defining the defaults. In section 2.7.4 (page 1482) there is description of default paragraph properties (in the docDefaults, pPr, pPrDefault, rPr, and rPrDefault nodes). When these XML nodes are not present *"the defaults are therefore application-defined"*.

The net result is that many customers will demand compatibility with defacto standard behaviours rather than ISO standards. Other vendors cannot achieve compatibility through OOXML as specified.

2. Spreadsheet Formats

Spreadsheets hold tables of data that are manipulated and arranged in rows and columns. These data entries can be either raw data, or formulas for calculating relationships between fields.

People use spreadsheets for many purposes ranging from note-taking, basic accounting, and (strangely enough) elections⁶.

2.1. Equations

Mathematical equations are typically stored in formats according to their underlying mathematical structure, rather than as visual graphics. This allows equations to be edited and interpreted based on their underlying mathematical structure.

Section 6.1.2.19 page. 5436 defines the "equationxml" attribute of "shape" elements, "used to rehydrate an equation using the Office Open XML Math syntax" [...] the "actual format of the contents of this attribute are application-defined", which makes them impossible to exchange between applications. Even though "they shall contain Office Open XML Math", this could be arbitrarily and unnecessarily obfuscated by the presence of other application-specific information, application-specific encodings, and so on.

<http://urltea.com/w0o> [140KB, HTML]

As mathematical formulas are defined in an undisclosed format, other vendors cannot correctly process formulas using OOXML.

2.2. Mathematical functions

2.2.1. Unit functions

When comparing units of measurement the CONVERT function is of use.

The CONVERT function (Part 4, Section 3.17.7.48) converts from one unit to another. Some conversions explicitly allowed include liquid measure conversions such as from liters to cups or tablespoons. But whose cup and whose tablespoon? Traditional liquid measures vary from country to country. In the US, a cup is 8oz, except for FDA labeling purposes when a cup is 240ml. But in Australia a cup is 250ml and in the UK it is 285ml. Similarly a tablespoon has various definitions. OOXML is silent on what assumptions an application should make.

<http://urltea.com/zik?rob-weir-blog> [76KB, HTML]

As OOXML doesn't define which units it's based upon other vendors cannot reliably implement

⁶ <http://urltea.com/zij?guardian-newspaper> [42KB, HTML]

OOXML. For example, does a “cup” change size by each computer's country settings, or is it static and if so is it strictly an FDA standard 240ml? Without this necessary information other vendors cannot provide accurate calculations and results to their customers.

2.2.2. Statistical functions

A confidence interval is used to express the reliability of an estimate through the use of an estimated range of values. OOXML has a “CONFIDENCE” function, however it is not fully defined.

It is hard to have confidence in the CONFIDENCE function (Part 4, Section 3.17.7.47). It is said to return the confidence interval around a sample mean given an alpha value, a standard deviation and a sample size. The problem is that this problem is under-defined. One must make an assumption, not stated here, as to the shape of the data distribution. Is it normally distributed data? Exponentially distributed? Weibull distribution? The standard does not define the meaning of this function sufficiently for one to implement it.

<http://urltea.com/zik?rob-weir-blog> [76KB, HTML]

OOXML does not define the statistical data distribution and therefore other vendors can only guess as to how this function is to be implemented. This may not be an easily understood function as it may have quirks. For example, OOXML requires vendors to implement dates that conflict with the Gregorian Calendar⁷ and without an accurate definition other vendors would not know whether this function too had quirks that must be emulated.

⁷ Ecma 376 section 3.17.4.1 page 3305, “Date Representation”, conflicts with the Gregorian calendar. See <http://urltea.com/11g3?ooxml-and-the-gregorian-calendar>

3. Presentation Formats

Presentation software is used to display slideshows of information to people. These can range from static slides to complex animation. Increasingly presentation software is being used for documentary movies. For example, the majority of Al Gore's *An Inconvenient Truth* was produced in Apple Keynote⁸.

3.1. Smart Tags

Smart Tags attach menus to words or phrases as they're typed in a document or presentation. For example, when the word "Microsoft" appears in a presentation a smart tag could create a menu linking to the companys website, or when a particular product is typed a menu would be created with the current price of this product. OOXML has a reference to smart tags however there is no formal definition or specification. As panelists from the British Standards Institute say:

smartTags element specifies existence of Smart Tags, but no reference to what these constructs are, or where to find a definition of them elsewhere in the standard.

<http://urltea.com/11cf?bsi-smarttags>

Here OOXML again relies on undisclosed information, and without further specification other vendors cannot reliably implement Smart Tags.

3.2. Browser Support

Most presentation applications include a web export for viewing presentations in web browsers and to assist search engine indexing. OOXML allows export of HTML targeted for 3 classes of browsers however these 3 options are at least ten years old (from 1997). This list does not reflect the reality of modern browsers, and as panelists from The British Standards Institute state:

[t]he restricted list of values provided in the list of supportable browsers, which only includes IE3, IE4 and Netscape3 and Netscape4, is totally unacceptable for an international standard.

<http://urltea.com/11cg?bsi-browsers>

Vendors such as Apple (with their own Office suite⁹) and The Mozilla Foundation¹⁰ would therefore be restricted from supporting their "Safari" and "Mozilla Firefox" browsers in OOXML. More to the point, they would be restricted from supporting W3C web standards such as HTML¹¹ and CSS¹².

8 <http://urltea.com/11jo?apple-keynote-movies>

9 <http://www.apple.com/iwork/>

10 <http://www.mozilla.org/foundation/>

11 <http://www.w3.org/TR/xhtml1/>

12 <http://www.w3.org/Style/CSS/>

Conclusion

We applaud Microsoft for their stated intent to standardise their office file format. However due to a number of application specific and undisclosed behaviours in the OOXML specification it is impossible for vendors other than Microsoft to fully implement OOXML. While Microsoft has granted patent use over the required portions of the specification that are described in detail the numerous undisclosed behaviours and inexplicit definitions are not covered, providing a legal as well as technical barrier to OOXML's implementation.

Until these technical and legal issues are resolved, the potential benefits of standardisation to consumers – such as interoperability – will not be realised.

If the proposed standard were to be approved as it stands OOXML would have all the formal trappings of a standard without the actual substance of which an open standard provides, and though it might confer an aura of standardisation on these file formats this appearance would be an illusion, and this gap between appearance and reality would be evidence of a failure of the standards process.

The standards process now needs to allow for the involvement of the wider document community to ensure that these issues are resolved, before the standard can proceed. Microsoft's OOXML was developed in private without industry involvement, and we suggest that Microsoft must now involve the wider document community in any future versions of OOXML in order to design truly open standards.

Full Disclosure

Matthew Cruickshank is a developer with some ten years experience using SGML and XML languages. He has been involved in the working group on the E-government's Web Guidelines, various Disability and Accessibility initiatives, and also the XML use in the e-GIF standard. Matthew develops open source conversion software called Docvert (<http://docvert.org/>) which is based on DocBook XML and ODF.

Chris Daish is a business development manager at Catalyst IT. He has a background in print and electronic publishing, and has latterly been involved in the writing on v2 of the New Zealand E-government Web Guidelines. He is currently Government Officer for the New Zealand Open Source Society (www.nzoss.org.nz), site editor of Open4Business (www.o4b.co.nz) and organiser of the New Zealand Open Source Awards (www.nzosa.org.nz).

Conal Tuohy is the senior software developer at the New Zealand Electronic Text Centre (www.nzetc.org), a digital library unit within the Library of Victoria University of Wellington. He has many years of experience developing software solutions using Microsoft office applications as well as with open XML technologies. He is currently a member of the Technical Council of the Text Encoding Initiative.

Disclaimer

The information contained in this document represents the current view of its authors. It does not (and cannot) cover future versions of OOXML or any future Microsoft Office format.

This document is not legal advice. We suggest that you consult a lawyer for that.

Appendix A: Other OOXML Analysis

This white paper researches whether the OOXML format is defined and whether patents have been granted, not whether OOXML has potentially other issues relating to ease-of-implementation, reuse of existing standards, flexibility, and good design, *etc.*

Vendors or indeed anyone researching OOXML may wish to consider these other resources,

- “Groklaw’s OOXML Objections” <http://urltea.com/v9y>
- “Six questions to national standardisation bodies” <http://urltea.com/11zx>
- “Dual Standards: More Choice, or Less?” <http://urltea.com/11zt> [280KB, PDF]
- “The Technical Case Against OOXML” <http://urltea.com/v9w> [240KB, PDF]
- “Rob Weir: The Case Against OOXML” <http://urltea.com/124r> [170KB, PDF]
- “When is a standard not a standard?” <http://urltea.com/12d7>
- “Google Inc’s Analysis of OOXML” <http://urltea.com/1200> [290KB, PDF]
- “British Standards Institute: BSI panelist comments” <http://urltea.com/11zy>
- “Arguments and Technical Comments” <http://www.nooxml.org/arguments>