The Case Against OOXML

This paper argues why DIS 29500 "Office Open XML" (OOXML) does not meet the criteria defined by ISO and others for an International Standard. This paper examines a small selection of several hundred specific serious flaws we have documented in OOXML.

1. Criteria for the Evaluation of Standards

What is a standard? Several relevant definitions are available. ISO says:

> "[A] document, established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context
>
> NOTE Standards should be based on the consolidated results of science, technology and experience, and aimed at the promotion of optimum community benefits."[1]

BSI British Standards says:

> "... a standard is an agreed, repeatable way of doing something. It is a published document that contains a technical specification or other precise criteria designed to be used consistently as a rule, guideline, or definition. Standards help to make life simpler and to increase the reliability and the effectiveness of many goods and services we use. They are intended to be aspirational - a summary of good and best practice rather than general practice. Standards are created by bringing together the experience and expertise of all interested parties such as the producers, sellers, buyers, users and regulators of a particular material, product, process or service."[2]

ISO/IEC JTC1 Directives say:

> "A purpose of IT standardization is to ensure that products available in the marketplace have characteristics of interoperability, portability and cultural and linguistic adaptability. Therefore, standards which are developed shall reflect the requirements of the following Common Strategic Characteristics:
>
> - Interoperability;
>
> - Portability;

---

1   ISO/IEC Guide 2:2004, definition 3.2. Several national standards boards have also adopted this ISO definition, e.g., Germany's DIN.
2   http://www.bsi-global.com/en/Standards-and-Publications/About-standards/What-is-a-standard/

- Cultural and linguistic adaptability."[3]

From these and other national definitions, some common themes emerge on what standards should do:

1. They define precise common criteria for doing something in a repeatable way.

2. They provide an optimal degree of order in a given context, intended to be aspirational, giving the consolidated results of science, technology and experience, a summary of good and best practice rather than general practice.

3. They encourage interoperability and portability.

4. They adapt to different cultures and languages.

This paper evaluates DIS 29500 "Office Open XML" (OOXML) against each of these criteria. Some specific examples of problems from the OOXML specification are given, but note that these are merely a handful of examples from a larger list of hundreds. The sheer volume of serious problems with OOXML demonstrates its immaturity as a specification and lack of suitability for Fast Track approval as an ISO standard.

2. Precise, Repeatable, Common

These criteria speak to the need for a standard to provide a detailed, written description that allows for the common practice of the technology.

First, the WordProcessingML part of OOXML lists a large number of "Compatibility Settings"[4] which provide Microsoft the ability to store information related to various behaviors from their legacy applications. These settings have names like: "footnoteLayoutLikeWW8", "autoSpaceLikeWord95" and "useWord97LineBreakRules."[5] However, the OOXML specification merely lists the names of these settings. It does not define them. Microsoft alone knows what these settings mean, but it declines to give a precise definition of them. Instead, OOXML refers the reader to legacy software applications:

> "To 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. If applications wish to match this behavior, they must utilize and duplicate the output of those applications."

---

3   JTC1 Directives, 5th Edition, Version 3.0, Section 1.2

4   Part 4, Section 2.15.3.9   All OOXML section references are from Ecma 376 "Office Open XML" specification, available at http://www.ecma-international.org/publications/standards/Ecma-376.htm

5   Other examples include: lineWrapLikeWord6, mwSmallCaps, shapeLayoutLikeWW8, supressTopSpacingWP, truncateFontHeightsLikeWP6, useWord2002TableStyleRules, wpJustification and wpSpaceWidth

This clearly is not precise and certainly does not provide for repeatable or common practice of these features. 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"[6] (the goal of OOXML according to its authors) is consequently reserved for Microsoft alone. The OOXML standard does not provide for repeatable or common practice of this benefit.

Second, the WordProcessingML part of OOXML lists a large number of list styles representing various different writing systems, language and business conventions.[7] These are given names such as "chicago", "ideographDigital", "ideographLegalTraditional", koreanDigital2" and "koreanLegal". These are merely labels, and again, are not precisely defined . The would-be implementors of the OOXML specification are told that something called "Korean Legal Numbering" exists, but they are not told what it means or how to practice it in their application.

For example, a would-be implementor of OOXML in Korea would be perplexed by a numbering style that merely says, "...the sequence shall consist of characters as defined in the Chicago Manual of Style" without specifying an edition of that manual (there have been 15 editions of *The Chicago Manual of Style*) or a page reference. The OOXML specification simply does not provide for repeatable, common use of these features.

Third, the SpreadsheetML part of OOXML describes a "securityDescriptor" attribute, which according to the specification[8]:

> "...defines user accounts who may edit this range without providing a password to access
> the range. Removing this attribute shall remove all permissions granted or denied to users
> for this range."

This is an important security-related feature that tells the application which users are allowed to edit a range in a spreadsheet without a password. A would-be programmer implementing this feature would need to know how these user accounts are represented in the document. Are they comma-delimited? Semi-colon delimited? Space-delimited? OOXML does not provide those details (although it does imply that more than one name is allowed). Also, there is no universal concept of digital identity. We all have multiple user accounts, for email, for database, for machine access, for domain controllers, for LDAP, etc. Which one is intended here? This function lacks sufficient definition to allow interoperability, which in the end is what repeatable, common use is all about.

In summary, many areas of OOXML are undefined or under-defined Although the specification does provide a formidable framework for Microsoft to represent its own documents in, this ability does not translate into anything approaching equal access for others to obtain these same benefits. The question to ask is, "Does OOXML define a document format in a precise way that allows repeatable and common practice of its claimed benefits?" The three examples above, and many others, demonstrate

---

6   Part 1, Introduction
7   Part 4, Section 2.18.66
8   Part 4, Section 3.3.1.69

that OOXML fails to satisfy this criterion. Its lack of maturity as a standard, reflected also in the lack of multiple full-featured implementations, and insufficient prior technical review, make it inappropriate for Fast Track consideration and a poor candidate for ratification as an International Standard.

3. Aspirational, Consolidated Best Practices

An ISO Standard should not merely be the minutely detailed record of the operating characteristics of a single company's product, no matter how dominant that company is in its field. From the definitions provided by ISO and others, cited earlier, an International Standard should represent the "consolidated results of science, technology and industry". A standard should be "aspirational." In other words, it should not just show one vendor's way of accomplishing a task. It should attempt to provide "a summary of good and best practice" based on the consensus of expert opinion. It should teach the best practices for the repeatable, common practice of a given technology.

Industry records its best practices through standardization. The existing body of document and markup standards represents a compendium of reviewed, approved, and implemented best practices. The work of the Word Wide Web Consortium (W3C)[9] is especially relevant to XML document formats, since they maintain the core XML standard as well as related standards such as XHTML, CSS2, XSL, XPath, XForms, SVG, MathML and SOAP, the standards that represent the very backbone of XML and XML-related technologies.

OOXML, however, incorporates very little of the consolidated best practices of the industry. Worse, would-be implementors of OOXML are asked to use Microsoft's proprietary, legacy formats, even when relevant and superior W3C standards are at hand.

For example, Vector Markup Language (VML) was developed by Microsoft and proposed by it to the W3C, where it was evaluated by a technical committee and rejected back in 1998. The industry instead supported Scalable Vector Graphics (SVG) which was developed into a standard by the W3C and then widely adopted. The standard for XML vector graphics has been SVG for almost a decade.   But OOXML uses the proprietary VML, because Microsoft integrated its proprietary VML rather than standard SVG into its Internet Explorer and Office 2000.

Microsoft has acknowledged that VML is the wrong standard to use for vector graphics:

> "The VML format is a legacy format originally introduced with Office 2000 and is included and fully defined in this Standard for backwards compatibility reasons. The DrawingML format is a newer and richer format created with the goal of eventually replacing any uses of VML in the Office Open XML formats. VML should be considered a deprecated format included in Office Open XML for legacy reasons only and new applications that need a file format for drawings are strongly encouraged to use preferentially DrawingML"[10]

---

9   http://www.w3.org
10 Part 4, Section 6.1

Instead of using the existing standard SVG, Microsoft OOXML includes two different markup languages for vector graphics, one that was rejected in 1998 by the W3C, and one that it developed in isolation.   The amount of extra work this causes for everyone who wishes to implement OOXML is immense. Implementors will need to support two different markups for the same function (neither of them standard) even though this gives no additional benefit to their users. Microsoft alone would benefit, since they have preexisting support for VML in Office.

Further, even more so than text, vector graphic are unlikely to be converted perfectly by file format translators. So the proliferation of redundant standards for vector graphics – two of them within OOXML – will lead to fidelity problems during conversions.

Does this sound aspirational? Does this sound as though it fosters best practices? On the contrary, 600 pages of VML requirements have been added to the OOXML specification that bring no value to anyone but Microsoft, and in fact creates steep barriers to others who would implement OOXML.

As a second example, note the definition of spreadsheet dates, where the following requirement is given:

> "For legacy reasons, an implementation using the 1900 date base system shall treat 1900 as
> though it was a leap year... A consequence of this is that for dates between January 1 and
> February 28, WEEKDAY shall return a value for the day immediately prior to the correct
> day, so that the (non-existent) date February 29 has a day-of-the-week that immediately
> follows that of February 28, and immediately precedes that of March 1."[11]

In other words, the Gregorian Calendar, the base calendar of commerce, science and government worldwide, is set aside for "legacy reasons." The result is that all would-be implementors of OOXML are required to have their applications give their users incorrect answers to questions like "What day of the week is February 1st, 1900?", if they want to conform to the OOXML standard. This causes particular pain in the common task of exchanging spreadsheet data with relational databases via SQL, a standard that explicitly requires the use of the Gregorian calendar.[12]

For a third example, note that OOXML defines a new string type called "Basic String" as "a binary basic string variant type."[13] One of the properties of this new string type is that it allows non-XML characters (control characters) to be specially encoded. However, the presence of non-XML characters in an XML document, breaks interoperability of XML and XML-based tools. The W3C's Internationalization Activity confirms this interpretation, saying:

> "Control codes should be replaced with appropriate markup. Since XML provides a
> standard way of encoding structured data, representing control codes other than as markup
> would undo the actual advantages of using XML. Use of control codes in HTML and
> XHTML is never appropriate, since these markup languages are for representing text, not

---

11 Part 4, Section 3.17.4.1
12 Database Language SQL—Part 2: Foundation (ISO/IEC 9075-2:1999), Section 4.7.3
13 Part 4, Section 7.4.2.4

data."[14]

Fourth, in several places[15] OOXML makes use of "bitmasks" to encode multiple boolean (true/false) values into a single integer. Although this was once common 20 years ago when programming in C in constrained memory environments, it is considered very bad style in XML. It makes processing by standard XML tools like XSLT extremely difficult, since these tools lack bit-level operations needed to effectively process data at the bit level.

Fifth, not only does OOXML fail to provide a consolidation of best practices from science, industry and experience, it fails to provide a consolidation of Microsoft's own best practices. OOXML recommends that print settings (number of pages to print, which pages to print, orientation, print quality, etc.) be stored in a platform-specific binary format. For example on Windows their guidance is to store in what is called the "DEVMODE" structure.[16] Doing so would render the print settings platform dependent and prevent interoperability. But at the same time, Microsoft's new specification, "XML Paper Specification" (XPS) offers a PrintTicket element of which Microsoft says:

> "PrintTicket technology is the successor of the current DEVMODE structure. It is an eXtensible Markup Language based document that specifies and persists information about job formatting and print job configuration.... Relative to the current print subsystem, the PrintTicket technology enables all components and clients of the print subsystem to have transparent access to the information currently stored in the public and private portions of the DEVMODE structure, using a well-defined XML format."[17]

Why is OOXML getting the inferior, binary, unportable, platform- and application-dependent print settings, when Microsoft's own recommended practice is to move to a "well-defined XML format?"

As a sixth example, note that OOXML defines several cryptographic algorithms[18] which are non standard. Instead of using an ISO/IEC 10118-3:2004 algorithm, or one approved for use by NIST in their FIPS-180 list of compliant algorithms[19] (and there are several on both lists, such as SHA-256), OOXML specifies a legacy hashing algorithm, presumably one used in earlier versions of Microsoft Office. Does this teach the consolidated best practices of science, industry and experience? On the contrary, Microsoft doesn't even recommend using these algorithms. Instead, they provide DRM-based protections in Office 2007 as undocumented extensions to OOXML. Since this DRM is not documented, no other vendor is able to freely use those features. Documents encrypted in Office 2007 cannot be read anywhere else. Would-be OOXML implementors instead have only the flawed legacy security support of OOXML, support which is not even FIPS-180 compliant. Again, Microsoft is keeping best practices to itself, and leaving the OOXML specification with crippled security.

In summary, OOXML is a direct port of a single vendor's binary document formats. The avoidance of

---

14 http://www.w3.org/International/questions/qa-controls
15 For example, Part 4, Section 2.3.1.6; Part 4, Section 2.4.51; Part 4, Section 2.4.52; Part 4, Section 2.4.7, etc.
16 Part 1, Section 15.2.14
17 http://msdn2.microsoft.com/en-us/library/ms715246.aspx
18 For example, in Part 4, Section 2.15.1.28
19 http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf

re-using relevant existing international standards, as well as the inconsistent use of Microsoft's own preferred technologies demonstrates that OOXML does not represent the consolidated results of science, industry and experience. It is not aspirational. Although it may provide a technique of reading data in that one vendor's format, that at best recommends it as only a technical specification. Since it does not represent the consolidated best practices in the industry, a defining quality of an ISO standard, the OOXML specification should not be approved as an International Standard.

4. Interoperable & Portable

Portability and Interoperability are two of JTC1's "Common Strategic Characteristics"[20] and as such are requirements of all JTC1-approved standards. In the realm of document format standards, the question is whether the proposed OOXML specification can be fully implemented by multiple applications on multiple operating systems. Or, has it been written exclusively for the benefit of a single vendor's application?

First, an important area of interoperability is the interchange of data between spreadsheets and relational databases. Many business processes are defined around this capability, which has been supported by most spreadsheet vendors for over a decade. However, OOXML has no way to represent dates before the year 1900, while modern databases can represent much earlier years. IBM's DB2 can support dates to the year 1, for example. Oracle supports dates back to the year 4712 B.C. The OOXML specification should not prevent any would-be implementors using dates as far back as they would wish. An application vendor will naturally want to match their spreadsheet's date support to the equivalent capabilities of their database. Why is OOXML restricted to the limitations of Microsoft Excel? This hurts interoperability between spreadsheets and databases.

Second, OOXML defines a ST_CF type[21], which records the allowed clipboard formats which may be used with a graphical object. The allowed values of this type, EMF, WMF, etc., are all proprietary Windows formats.   No allowance has been made for use by other operating systems. For example, in Linux images are typically copied on the clipboard in an open standard format like PNG. But if a vendor encodes "PNG" into a document record of this type, the document will be invalid, and the document and the application will not conform to the OOXML specification.

Third, the definition of a password hashing algorithm in SpreadsheetML is given by presenting 5-pages of C-language source code[22], likely extracted from Excel. However, the bit manipulations of this code are inherently machine-dependent, and will give different results depending on the processor architecture. A document created on one machine may not be readable on a different machine. OOXML has not provided a portable definition of this function.

Fourth, the "optimizeForBrowser" element of WordProcessingML[23] has been defined in a way which

---

20 JTC1 Directives, 5th Edition, Version 3.0, Section 1.2
21 Part 4, Section 6.4.3.1
22 Part 4, Section 3.2.29, pg. 1917
23 Part 4, Section 2.15.2.32

ignores the existence of current browsers other than Internet Explorer. What about Firefox? What about Safari? What about Opera? None of these can be set as target browsers. This section in OOXML requires that "all settings which are not compatible with the target web browser shall be disabled." What if I want my application to produce standards-compliant output? So yes to PNG, no to VML, yes to MathML and SVG? A would-be implementor is not able to specify this with the way OOXML has been designed.

Sixth, the "Slide Synchronization Properties" feature of DrawingML.[24] provides the ability for a presentation to synchronize slide content with centrally-stored slides on a server. This is a feature of Microsoft PowerPoint and SharePoint. However, the description of this feature in OOXML lacks sufficient details. What is the communication protocol? What is the data model? Although standards exist for describing a client-server protocol of this sort, namely the various Web Services standards, OOXML gives no information. Independent interoperable implementations of this function are prevented and the one implementation that exists will be tied to SharePoint.

In Summary, where OOXML references other technologies it often does so in a way that ties it exclusively to the technologies already supported by Microsoft Office. In some cases extraordinary efforts are made to incorporate other specifications, like VML, into OOXML. Not only does OOXML ignore alternative, standard and open technologies, it prevents other vendors from adding interoperable support for other technologies. Although any vendor is entitled to their own design decisions and their own priorities, an ISO standard must have the characteristics of portability and interoperability, so that all vendors may have that same right to their own design decisions and priorities. The arbitrary restrictions of OOXML, which work extremely well with Microsoft's solutions and platforms, but not others, render the proposed specification unsuitable for approval as an International Standard.

5. Cultural & Linguistic Adaptability

Since OOXML's features derive from the feature set of Microsoft Office, it is not surprising that this feature set best reflects the needs of developed countries and communities where Microsoft's business has seen the greatest success. However, an International Standard must take a broader view and provide wide cultural and linguistic interoperability.

An example of a concern is the spreadsheet function NETWORKDAYS()[25]. This function is defined by OOXML to return the number of working days between two dates, exclusive of any weekends in that interval. For some cultures, the weekend is Saturday and Sunday. For others, the days of rest are either Thursday/Friday or Friday/Saturday.   OOXML does not define "weekend" and does not provide a way for the user to define it either. As implemented in Excel the function assumes the weekend is always Saturday/Sunday. This spreadsheet function is defined in a way which renders an incorrect answer for potentially billions of people across the globe. OOXML lacks cultural adaptability. Compare this to the same function in OpenDocument Format, where the user may pass in an additional parameter to override the default definition of a weekend.

---

24 Part 4, Section 4.7.1
25 Part 4, Section 3.17.7.224

Second, WordProcessingML has a feature called "Border Styles"[26] which lists a large number of graphical borders which can be used as page borders. These represent a closed list of specific named border styles with mandated images. An example of two such graphics is shown in Figure 1.
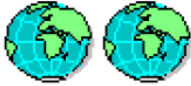
earth1 (Earth Art Border)	Specifies an art border consisting of a repeated image of Earth, as follows (showing two repetitions):
earth2 (Earth Art Border)	Specifies an art border consisting of a repeated image of Earth, as follows (showing two repetitions):

Figure 1:
Illustration 1: Page Borders

These are the only two possibilities for displaying a globe in a page border and neither of them show Asia. Similarly, there are graphics for birthday cakes, St. Valentine's Day cupids, painted Easter eggs, Christmas gingerbread men, Halloween Jack O'Lanterns, and other images that are appropriate for a Western cultural milieu, but have limited application elsewhere. The problems here are that this list of page border styles is a closed list and it matches exactly what Microsoft Word provides. A would-be implementor of OOXML may not extend this list with additional images types to better suit the cultural milieu of their customers. If they do, their documents will not be valid OOXML and the application that allows non-standard images to be used as page borders will not conform to the OOXML specification. How well does OOXML adapt to other cultures? In the case of page borders, it fails to provide adaptability.

Third, as mentioned previously, WordProcessingML defines a number of numeration styles for numbered lists.[27] These numeration styles were essentially only labeled, but not defined. These styles are also defined as a closed list, again matching what Microsoft Word supports, but they are not extensible by other vendors. However, the list of styles provided is incomplete, lacking, for example, support for Armenian, Tamil, Greek alphabetic, Ethiopic and Khmer numerations, as well as the larger number of historic systems used by scholars. The preferred solution is to use a declarative/generative approach, such as used by the W3C's XSL:FO and OpenDocument Format. This allows an open-ended list of numeration styles to be used, each self-defining.

---

26 Part 4, Section 2.18.4
27 Part 4, Section 2.18.66

Cultural and linguistic adaptability suffers in OOXML because of closed-ended lists which, although they may match perfectly what Microsoft Office offers today, are not extensible by vendors in an interoperable way.

6. Summary

Standards have standards. Evaluating the proposed OOXML specification based on the criteria provided by ISO for what a standard should be, this paper has detailed where OOXML failed to meet the various desired characteristics of ISO standards: precision, common criteria, optimal degree of order, being aspirational, consolidating the best practices of science, technology and experience, interoperability, portability and cultural and linguistic adaptability. By many examples we have shown that the proposed OOXML standard falls short of this mark. By failing to meet these criteria OOXML has failed to provide for the optimum community benefit. Indeed, the proposal appears to be targeted to benefit a single corporation only.

The expectations for a document format standard are high, and they should be. A standard document format that meets the above criteria is essential to long-term preservation of our digital heritage, for equal access to government documents and records by all citizens, and for cost-effective and efficient document-based business process integration and workflows across heterogeneous systems. OOXML, the file format for Microsoft Office, does not provide these benefits, and is not suitable for an ISO standard. JTC1 is urged to vote disapproval on this ballot.

— Based on contributions by Rob Weir at IBM and others.