51	Clause No. / Subclause No. / Annex (e.g. 3.1)	Paragraph / Figure / Table / Note (e.g. Table 1)	Type of comment (ge=general, te=technical, ed=editorial)	Comment (justification for change) by the MB	Proposed change by the MB
	Office Open XML Overview	-	ge	This document is not listed as part of the Ecma 376 standard in the Forward to Part I “Fundamentals” and its status whether informative or normative is not explicitly stated.	Clarify the status of this Overview document. If it is merely a promotional whitepaper about Ecma 376, then it should not be included in the published standard.
1	Office Open XML Overview	§1, p. 1	ge	According to the text, the proposed standard "was designed from the start to be capable of faithfully representing the pre-existing corpus of word-processing documents, presentations, and spreadsheets that are encoded in binary formats defined by Microsoft Corporation". Such a design goal should not take place in the standardization effort. While the office software tools should support the legacy formats as well as the new standards, the new standards should not be encumbered with all the deprecated features of the legacy formats.	Remove the reference to the pre-existing Microsoft formats
1	Part 1, Appendix		te	The reference given for the Zip format does not provide a date or version, though this specification is frequently revised,	Reference should be made to a particular dated and labeled version.
1	Part 1, Section 8.6.2	§8.6.2	te	VML is presented as deprecated and included in the standard for backwards compatibility reasons only, while new applications are strongly encouraged to use preferentially DrawingML. This recommendation contradicts Part 3 §2.3.1 which says "All background information in a WordprocessingML document is stored using the VML syntax".	Clarify this contradiction
1	Part 1, Section 10.1.2	line 20	te	Reference is made to material in Part 5, Clause 12. Although a clause of that number does exist, it does not contain the material 10.1.2 references it for. In addition, this clause is not normative.	Correct the reference to point to the correct clause.
1	Part 1, Section 12.3.5	-	te	This binary part is said to be used for the storage of “arbitrary user-defined data”. No further detail is given as to what user action would trigger the use of this “user-defined” data. Without further definition, no interoperability of this feature is possible.	Fully define the use of Custom Property Part
1	Part 1, Section 15.2.12	-	te	There is no reference made to a particular dated version of TrueType or OpenType specifications. And if TrueType and OpenType differ, then there should be different ways to refer to them, rather than calling them both “application/x-font-ttf”	Provide reference to intended specifications for TrueType and OpenType
1	Part 1, Section 15.2.14	-	te	It is unsatisfactory to store printer settings in OS-dependent binary formats like DEVMODE structures. This is a considerable security concern (DEVMODE structures are loaded directly into device driver memory), as well as lacking cross-platform adaptability. There is also no interoperability with print settings as currently defined.	Alternatives are available for expressing print settings in XML rather than in binary. For example, Microsoft's own XPS specification defines a PrintTicket markup for which the XPS specifications says, “The PrintTicket is XML that provides print settings in a consistent, accessible, and extensible manner. We would like the same qualities in OOXML's print settings, not a binary blob.
	Part 1, Section 15.2.6	-	te	What is meant by “This part shall have no contents”? Does this mean that there shall be nothing in the Zip file with the declared name? Or does it mean that a zero-byte file shall be created with the declared name? Or something else?	Clarify the meaning.
	Part 1, Section 2.4	line 22	te	This line require conformance with “Unicode Standard” without specifying a version. XML 1.0 referred to Unicode 2.0, though the informative Appendix A of OOXML Part 1 lists Unicode 4.0. Which is it?	An explicit Unicode version reference should be made in the Conformance section.
	Part 1, Section 9.1.1	-	te	ASCII requires a normative reference since there are several national variations.	Suggest reference be made to ISO/IEC 646:1983 or ANSI X3.4-1986
1	Part 1, Section 9.1.5	-	te	This sub-clause, buried in introductory material, negates a provision of the more detailed OPC specification in Part 2. This will likely be missed by implementors.	If interleaving is not permitted then it should not be described in Part 2.
	Part 4, Introduction	page vii, line 8	te	An XML markup cannot be “fully compatible” with an “investment”	Remove the fluff
	Part 4, Section 2.15.1.28	-	te	This says that document protection “shall be enforced”. “Shall” indicates required behavior. But then a few sentences later it says that document protection “may be ignored”.	Clarify this contradiction.

	Part 4, Section 2.15.1.28	-	te	A hash algorithm is provided, likely based on a legacy algorithm used in Word. This legacy algorithm is known to be a weak algorithm and has effectively been cracked. One could argue that no hash algorithm would be effective in OOXML, since a user could simply unzip the document and hand edit the XML to remove the hash or to set it to some known value. However, some application types such as online editing via Google Docs, or other similar applications, can secure physical access to the document via other means. Editing access to the document does not necessarily presuppose physical access to the document's XML. So there is a necessity for a secure & interoperable hash algorithm, such as SHA-256 for document protection.	Use a standard, FIPS-180 compliant hash algorithm as the default. Legacy hash algorithms should be supported via the described extension mechanism.
	Part 4, Section 2.15.1.28	line 13	te	This algorithm description fails to specify the encoding of the input password. Presumably it is Unicode, but in what encoding? UTF-16BE? UTF-16LE? UTF-16 with a BOM (Byte Ordering Mark)? The described algorithms make use of byte-level manipulations which depend on the machine architecture (big endian versus little endian). So it is necessary that all byte ordering assumptions be made explicit.	Make the byte ordering assumptions explicit, both for the input password and the processing steps, so as to allow cross-platform interoperability. Keep in mind that the hash may be calculated on a different machine architecture than the password was entered with.
	Part 4, Section 2.15.1.28	pg 1159, lines 6-9	te	The described processing steps are not ambiguous. In particular SHR and SHL give different results on different machines and with signed and unsigned values	Describe the hash algorithm in a platform independent manner.
1	Part 4, Section 2.15.1.86	-	te	This element uses a bitmask to specify a style display filter. The use of bitmasks rather than a set of boolean types makes this data almost impossible to work with standard XML tools like XSLT which lack bit-level operations.	Rewrite this subclause to express the feature using XML constructs rather than bitmasks.
	Part 4, Section 2.15.1.87	-	te	This element uses a bitmask to specify style display sorting parameters. The use of bitmasks rather than a set of boolean types makes this data almost impossible to work with standard XML tools like XSLT which lack bit-level operations.	Rewrite this subclause to express the feature using XML constructs rather than bitmasks.
1	Part 4, Section 2.15.2.32	-	te	This feature has been defined in a way which 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 requires that “all settings which are not compatible with the target web browser shall be disabled.” But what if I want my application to produce standards-compliant output? So yes to PNG, no to VML, yes to MathML and SVG? I can't seem to specify this.	Ecma should rethink the entire optimizeForBrowser subclause. It looks very much like it is mapping directly to the arbitrary choices of a single vendor's application. This clause should be rewritten to express this feature in an application and platform neutral way.
1	Part 4, Section 2.15.3	-	te	These “compatibility” settings solve no general problem. They are merely a museum of settings from previous versions of Microsoft Word. No allowance has been made for legacy settings from other applications. Better to have these be application-specific settings using the existing extensibility mechanisms of OOXML.	Remove the compatibility settings from OOXML.
	Part 4, Section 2.15.3.26	-	te	This is the “footnoteLayoutLikeWW8” element, which is defined in terms of mimicking a legacy application's behavior. The standard contains insufficient detail on how to replicate this behavior.	Define the intended behavior.
1	Part 4, Section 2.15.3.31	-	te	This is the “lineWrapLikeWord6” element, which is defined in terms of mimicking a legacy application's behavior. The standard contains insufficient detail on how to replicate this behavior.	Define the intended behavior.
	Part 4, Section 2.15.3.32	-	te	This is the “mwSmallCaps” element, which is defined in terms of mimicking a legacy application's behavior. The standard contains insufficient detail on how to replicate this behavior.	Define the intended behavior.
	Part 4, Section 2.15.3.41	-	te	This is the “shapeLayoutLikeWW8” element, which is defined in terms of mimicking a legacy application's behavior. The standard contains insufficient detail on how to replicate this behavior.	Define the intended behavior.
	Part 4, Section 2.15.3.51	-	te	This is the “suppressTopSpacingWP” element, which is defined in terms of mimicking a legacy application's behavior. The standard contains insufficient detail on how to replicate this behavior.	Define the intended behavior.

1	Part 4, Section 2.15.3.54	-	te	This is the " uiCompat97To2003" element, which is defined as: "Disable UI functionality that is not compatible with Word97-2003". But what use is this if I am using OOXML in OpenOffice or WordPerfect Office? What if I want to disable UI functionality that is not compatible with OpenOffice 1.5? Or WordPerfect 8? Or any other application? Where is the ability for other implementations to specify their preferences?	Define this an application neutral way. If it is truly a Word-only feature, then remove it from OOXML and express as an application-defined extension.
	Part 4, Section 2.15.3.54	-	te	This is the "truncateFontHeightsLikeWP6" element, which is defined in terms of mimicking a legacy application's behavior. The standard contains insufficient detail on how to replicate this behavior.	Define the intended behavior.
1	Part 4, Section 2.15.3.6	-	te	This is the "autoSpaceLikeWord95" element, which is defined in terms of mimicking a legacy application's behavior. The standard contains insufficient detail on how to replicate this behavior.	Define the intended behavior.
1	Part 4, Section 2.15.3.63	-	te	This is the "useWord2002TableStyleRules" element, which is defined in terms of mimicking a legacy application's behavior. The standard contains insufficient detail on how to replicate this behavior.	Define the intended behavior.
1	Part 4, Section 2.15.3.64	-	te	This is the "useWord97LineBreakRules" element, which is defined in terms of mimicking a legacy application's behavior. The standard contains insufficient detail on how to replicate this behavior.	Define the intended behavior.
1	Part 4, Section 2.15.3.65	-	te	This is the " wpJustification" element, which is defined in terms of mimicking a legacy application's behavior. The standard contains insufficient detail on how to replicate this behavior.	Define the intended behavior.
1	Part 4, Section 2.15.3.66	-	te	This is the "wpSpaceWidth" element, which is defined in terms of mimicking a legacy application's behavior. The standard contains insufficient detail on how to replicate this behavior.	Define the intended behavior.
1	Part 4, Section 2.16.5.33	-	te	This does not define how a picture is named. Is it by a URI? By a local file system path? Either? The example given has a DOS file path, a construct which is not portable.	Define how pictures are named.
	Part 4, Section 2.16.5.33	-	te	This subclause defines an INCLUDEPICTURE field which "Retrieves the picture contained in the document named". However, no mention is made of what formats are permissible for the picture.	There should be specified at least a small set of interoperable formats.
	Part 4, Section 2.16.5.34	-	te	This does not define how a document is named. Is it by a URI? By a local file system path? Either? The example given has a DOS file path, a construct which is not portable.	Define how documents are named.
1	Part 4, Section 2.16.5.34	-	te	This subclause defines an INCLUDETEXT field which "Inserts all or part of the text and graphics contained in the document named". However, no mention is made of what formats are permissible for the retrieved text.	There should be specified at least a small set of interoperable formats.
	Part 4, Section 2.16.5.34	-	te	The \t flag will apply a named XSLT transform to the input XML file and insert the resulting output. However, no proper reference is given to XSLT, so we do not know what version XSLT transform is permitted here.	Provide a proper external normative reference for the XSLT which is allowed here.
1	Part 4, Section 2.16.5.41	-	te	This describes a "MACROBUTTON" field which can run a designated macro or command. But there is no mention of what programming language or API's are allowed for such a designated macro or command.	Described this feature to a level where cross-platform, cross-application interoperability is possible.
	Part 4, Section 2.16.5.5	page 1512, lines 11-12	te	According to the text, the AUTONUM field is deprecated. A new standard should not contain deprecated parts.	Remove all references to AUTONUM from the OOXML text.
1	Part 4, Section 2.18.4	-	te	The artwork provided here is of poor quality providing neither intended scale, spacing, color depth, etc. A small example diagram is an insufficient definition. For example, are the dimensions of the borders absolute? Or do they scale with page size? Also, some of the images, such as 'apples' or 'balloons3Colors' show copying artifacts, where extraneous lines or blotches appear.	Provide full normative definitions for these graphical elements. Also, for informative purposes, these graphics may be provided in standalone file form, preferably in a scalable graphics format like SVG.
1	Part 4, Section 2.18.4	-	te	No mechanism for expanding the set of art borders is provided. Since the specified art borders are heavily Western-oriented, it would be good to provide a way for an application to supplement these styles with graphics that provide more regional flavor.	Provide an interoperable extensibility mechanism for a document author or application to specify their own art border graphics.
1	Part 4, Section 2.18.45	-	te	Length is said to be "exactly 3 characters". This is inconsistent with the example given which has a length of 6 characters.	Clarify the definition. In particular note that xsd:hexBinary measure length in octets, not characters.

1	Part 4, Section 2.18.51	-	te	The use of 255 enumerated language codes, in addition to ISO 639-1 codes, adds no expressive value and only increases the work required of any application that would process an OOXML document.	Drop the use of the redundant ST_LangCode
1	Part 4, Section 2.18.66	-	te	The formatting system described here is not comprehensive, lacking, for example, support for Armenian, Tamil, Greek alphabetic, Ethiopic and Khmer numerations, all in use today, as well as the various historical systems still used by scholars.	Use a more flexible, extensible, generative approach to numeration, such as that used by the W3C's XSLT standard in their xsl:number support
1	Part 4, Section 2.18.66	-	te	There is nothing in this section which is normatively defined except some enumeration values. No normative meanings to these values are given. An informative example is insufficient in all but the most trivial cases. For example, where is “Korean Legal Counting System” defined?	Give explicit definitions of these numbering styles or proper external normative references.
1	Part 4, Section 2.18.66	“chicago”	te	Format is defined in reference to the “Chicago Manual of Style”, but no edition or page reference is provided.	Either include the entire definition in the standard, or provide a proper external reference.
1	Part 4, Section 2.18.66	“decimalFullwidth”, etc.	te	There are several mentions of double-byte and single-byte Arabic numbering schemes. Since the conformance clause for OOXML requires the use of Unicode in UTF8 or ITF16 encodings, there should be no mention of other encodings.	Reconcile the text and the conformance clause..
	Part 4, Section 2.18.66	“lowerLetter”, etc.	te	Several counting systems are defined to use letters of the alphabet, but nothing is mentioned about how counting continues once the letters of the alphabet are exhausted.	Clarify the text to explicitly cover this case.
	Part 4, Section 2.18.66	“numberInDash”, etc.	te	Format requires use of “dash” to surround the number, but no indication of which Unicode dash is intended, en-dash, em-dash, hyphen-minus, figure-dash, quotation-dash, etc.	Specify the intended dash explicitly.
	Part 4, Section 2.18.72	-	te	No definition is provided for a “Panose-1 classification” of a font.	Provide a proper external normative reference for this term.
	Part 4, Section 2.18.85	-	te	The fill patterns lack definitions. The illustrations given are insufficient. An application needs to know what in these illustrations are required behaviors and what are not. For example, is the exact dithering pattern used in the illustration required?	Provide full normative definitions for these graphical elements.
	Part 4, Section 2.3.1.8	-	te	This element uses a bitmask to specify various paragraph conditional formatting properties.. The use of bitmasks rather than a set of boolean types makes this data almost impossible to work with standard XML tools like XSLT which lack bit-level operations.	Rewrite this subclause to express the feature using XML constructs rather than bitmasks.
1	Part 4, Section 2.3.3.19	-	te	This says that “The layout properties of this embedded object are specified using the VML syntax”. However, in Part 1, Section 8.2.6 says, “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” Certainly a new document creating an OLE embedding should not be using VML. Otherwise, all OOXML consumers will need to support VML, even where legacy documents are not present.	Define layout properties of embedded objects using DrawingML rather than VML
1	Part 4, Section 2.4.51	-	te	This element uses a bitmask to specify various table style formatting properties.. The use of bitmasks rather than a set of boolean types makes this data almost impossible to work with standard XML tools like XSLT which lack bit-level operations.	Rewrite this subclause to express the feature using XML constructs rather than bitmasks.
	Part 4, Section 2.4.52	-	te	This element uses a bitmask to specify various table style formatting exceptions. The use of bitmasks rather than a set of boolean types makes this data almost impossible to work with standard XML tools like XSLT which lack bit-level operations.	Rewrite this subclause to express the feature using XML constructs rather than bitmasks.
1	Part 4, Section 3.17.4.1	-	te	The restriction to only two date bases is arbitrary and based only on one vendor's applications. There are other reasonable values for date bases, including earlier ones for historical dates.	Allow a range of vendor-declared date bases, or explicitly allow negative date serial values to express dates prior to 1900
1	Part 4, Section 3.17.4.1	-	te	The mandated incorrect date calculations for 1900 in the 1900-based date basis is unacceptable. An ISO standard should not be mandating incorrect values for the well-established Gregorian Calendar. To do so will only lead to confusion, poor interoperability and perpetuation of errors.	If needed for legacy reasons with legacy Excel documents, then introduce an additional vendor-specific tag such as “doWrongDateCalculationsLikeExcel” or similar. This is the approach recommended elsewhere in OOXML for legacy Word features.
	Part 4, Section 3.17.4.1	page 2522, lines 14-18	te	The text proposes a dual date base system. There is no clear advantage to having two slightly different systems, and this brings significant costs and confusion, as illustrated by the need to specify a default base system, etc.	Choose and keep a single date system.

	Part 4, Section 3.17.4.1	page 2522, lines 16&18	te	The documented upper limits for serial date times match 9999-12-31 00:00:00, which is most probably not what was intended. The expected upper limits would match 9999-12-31 23:59:59.	Clarify the upper limits.
1	Part 4, Section 3.17.4.1	page 2522, lines 19	te	The proposed date system does not cope with dates prior to 1900-01-01.	Propose a better date system.
	Part 4, Section 3.17.7.341	-	te	As written this function mandates an incorrect calculation for day of week for certain dates in the year 1900. An ISO standard should not be mandating incorrect values for the well-established Gregorian Calendar. To do so will only lead to confusion, poor interoperability and perpetuation of errors.	Remove the text that defines behavior that results in incorrect date calculations.
	Part 4, Section 3.2.29	-		A hash algorithm is provided, likely based on a legacy algorithm used in Excel. This legacy algorithm is known to be a weak algorithm and has effectively been cracked. One could argue that no hash algorithm would be effective in OOXML, since a user could simply unzip the document and hand edit the XML to remove the hash or to set it to some known value. However, some application types such as online editing via Google Docs, or other similar applications, can secure physical access to the document via other means. Editing access to the document does not necessarily presuppose physical access to the document's XML. So there is a necessity for a secure & interoperable hash algorithm, such as SHA-256 for document protection.	Use a standard, FIPS-180 compliant hash algorithm as the default. Legacy hash algorithms should be supported via the described extension mechanism.
	Part 4, Section 3.2.29	p. 1917-1922	te	No normative description of the password hashing algorithm is provided, so interoperability of this feature cannot be assumed. In an informative section, 5-pages of C-language source code is provided as “an example”, and this appears to involve machine-dependent bit manipulations.	Provide a normative, cross-platform definition of the hashing algorithm. Cross-platform source code can be given as an example, but the normative text should be in English, not in a programming language.
	Part 4, Section 3.2.29	pg. 1916	te	This seems to imply that if a password is entered in a script like Armenian or Ethiopic then the characters will be replaced all by a single character 0x3F, making the protection feature useless. This is unacceptable.	Remedy so password hashes can be calculated on any Unicode password.
	Part 4, Section 3.2.29	pg. 1916	te	This algorithm description fails to specify the encoding of the input password. Presumably it is Unicode, but in what encoding? UTF-16BE? UTF-16LE? UTF-16 with a BOM (Byte Ordering Mark)? The described algorithms make use of byte-level manipulations which depend on the machine architecture (big endian versus little endian). So it is necessary that all byte ordering assumptions be made explicit.	Make the byte ordering assumptions explicit, both for the input password and the processing steps, so as to allow cross-platform interoperability. Keep in mind that the hash may be calculated on a different machine architecture than the password was entered with.
	Part 4, Section 3.2.29	pg. 1916	te	The conversion from input password to single byte string is ambiguous. Certainly the input password could contain characters from more than one script, say some Korean, some Chinese. Do we process via multiple DBCS code pages? Or just one and then replace the unmapped characters with 0x3F? If only one DBCS code page is used, how is that determined in this case?	Clarify this processing, especially for passwords that use characters from more than one script.
1	Part 4, Section 3.3.1.61	-	te	The pageSize attribute allows a set of enumerated values which does not encompass all of the page size values permitted by ISO 216, ANSI Y14.1 and similar DIN and JIS standards.	Rather than trying to maintain a paper size registry, a more flexible approach would be to simply record the dimensions of the paper size selected.
	Part 4, Section 3.3.1.69	-	te	No normative description of the password hashing algorithm is provided, so interoperability of this feature cannot be assumed. In an informative section, C-language source code is provided as “an example”, and this appears to involve machine-dependent bit manipulations.	Provide a normative, cross-platform definition of the hashing algorithm. Cross-platform source code can be given as an example, but the normative text should be in English, not in a programming language.
	Part 4, Section 3.3.1.69	-	te	The securityDescriptor attribute, “defines user accounts who may edit this range without providing a password to access the range”. It is a string. But no information is given as to what user accounts are referred to here, or what the delimiter is. Are these comma-delimited local machine user accounts? Or semi-colon delimited LDAP DN's? There will be no interoperability if this is not defined.	Fully define this attribute.

	Part 4, Section 3.3.1.69	-	te	A hash algorithm is provided, likely based on a legacy algorithm used in Excel. This legacy algorithm is known to be a weak algorithm and has effectively been cracked. One could argue that no hash algorithm would be effective in OOXML, since a user could simply unzip the document and hand edit the XML to remove the hash or to set it to some known value. However, some application types such as online editing via Google Docs, or other similar applications, can secure physical access to the document via other means. Editing access to the document does not necessarily presuppose physical access to the document's XML. So there is a necessity for a secure & interoperable hash algorithm, such as SHA-256 for document protection.	Use a standard, FIPS-180 compliant hash algorithm as the default. Legacy hash algorithms should be supported via the described extension mechanism.
	Part 4, Section 5.1.12.28	-	te	This type is used in only two places, 5.1.2.2.32 and 5.1.2.2.33, in both cases to represent an RGB color value. Since you already have defined a ST_HexColorRGB type that should be used.	Use the ST_HexColorRGB type and remove ST_HexBinary3
	Part 4, Section 5.1.12.37	-	te	No definition is provided for a “Panose setting of a font”.	Provide a proper external normative reference for this term.
	Part 4, Section 5.1.12.37	-	te	The Panose value is said to be used, “so that generating applications using this Office Open XML Standard may determine the closest font type if necessary”. However, no font distance metric or font matching heuristic is described.	Describe the intended font matching procedure.
1	Part 4, Section 5.1.12.37	-	ge	Why are there several different definitions for a Panose value, both in Word Processing ML as well as Drawing ML?	Since they are exactly the same they should be defined once in a shared schema.
1	Part 4, Section 5.1.3.4	-	te	This describes the attachment of a QuickTime video to a presentation object. No description of the QuickTime format is provided. Without specifying a version and supported codecs, there will be no interoperability.	Provide an external reference for the version(s) of QuickTime format intended here as well as an interoperable codec.
1	Part 4, Section 6	-	te	OOXML specifies here a markup language called Vector Markup Language (VML) which, in addition to DrawingML, specifies a vocabulary for describing graphical objects. Section 6.1 says, “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” The need to support VML by OOXML consumers, in addition to DrawingML, would come at great implementation expense (the VML specification is over 600 pages) , would disadvantage all vendors but Microsoft, and would hurt interoperability.	Remove VML from OOXML. Vendors who have access to the legacy binary format documentation, such as Microsoft, are free to convert the VML to the “newer and richer” DrawingML at the same time they convert the document to OOXML.
1	Part 4, Section 6.	pages 4343-4960	te	All subsections of Section 6 describe deprecated only material, making Section 6 deprecated as a whole. A new standard should not contain deprecated parts.	Remove Section 6.
1	Part 4, Section 6.1.2.19	pg. 4653 “equationxml”	te	This describes the "equationxml" attribute of "shape" elements, "used to rehydrate an equation using the Office Open XML Math syntax". However, the "actual format of the contents of this attribute are application-defined", which makes them impossible to exchange between applications. If we're going to have a new math markup language in XML, and ignore the existing MathML, let's at least use the new markup in its elemental form, as well-formed XML (not stuffed into an attribute value), and without extending it in application-dependent ways.	Define equations in an interoperable way.
1	Part 4, Section 6.1.2.19	pg. 4655, “gfxdata”	te	Describes 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. If we're going to have a new graphics markup language in XML, and ignore the existing SVG, let's at least use the new markup in its elemental form, as well-formed XML (not stuffed into an attribute value), and without extending it in application-dependent ways.	Define this in an interoperable way.

	Part 4, Section 6.2.2.14	-	te	This describes an "ink" element which stores "ink annotations in an application-defined format." This is apparently intended to store 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 OOXML. We are opposed to standardizing placeholder elements for entirely application-dependent proprietary formats without also specifying an interoperable format for those who with to create interoperable formats.	Specify the “ink” format or remove the element from OOXML and make this an application extension using the extensibility mechanisms of OOXML.
	Part 4, Section 6.4.2.10	-	te	This element is defined as providing a, “general-use element for objects that use an image representation, such as OLE objects, embedded controls, cameras and signature lines.” However, the allowed values, EMF, WMF, etc., refer to formats for which no reference has been given.	Provide a proper external normative reference for the allowed formats containable within this element.
1	Part 4, Section 6.4.3.1	-	te	The allowed values of this enumeration, EMF, WMF, etc., are Windows-specific formats. No allowance seems to have 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.	Several options here, but the desire is to allow cross platform interoperability.
1	Part 4, Section 7.1	-	te	This is the specification of Office Open Math Markup Language, a specialized XML vocabulary for the describing the layout of mathematical equations. This solves the same problem as MathML, a long-established W3C standard and an ongoing activity in the W3C. Since the equation editing feature of Word was entirely rewritten in Word 2007, there doesn't not seem to be the argument that an additional equation language must be introduced for the sake of legacy documents.	It is recommended that this section be removed from OOXML and that the proposers of OOXML work within the W3C's MathML activity, where MathML 3.0 is currently being drafted, to produce a single standard for equations that can be used later referenced by a future version of OOXML.
1	Part 4, Section 7.4.2.4	-	te	The presence of non-XML characters, escaped, or not escaped in an OOXML document, is contrary to 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 data.”	Remove the bstr type from OOXML
1	Part 4, Section 7.4.2.4	-	te	This defines a new XML string type which allows the inclusion via an escape mechanism of Unicode characters which are otherwise impermissible in XML documents. However, any escape mechanism must also specify a mechanism for “escaping the escape”. So, how does one represent the literal example given in 7.4.2.4 in a bstr?	Complete the definition of the escape mechanism.
1	Part 4, Section 7.4.2.5	-	te	It doesn't make sense for us to be specifying strings as null-terminated C-style strings and then to base-64 encode that. That is avoiding XML and will cause the markup to interoperate poorly with XML-based tools.	Ecma should rethink the entire Clipboard Data representation. It looks very much like it is mapping directly to the arbitrary internals of a single application. This clause should be rewritten to express this feature in an application and platform neutral way.
1	Part 4, Section 7.4.2.5	-	te	The value of -3 specifies a GUID that contains a format identifier (FMTID). The required format for neither a GUID nor a FMTID is specified.	Specify this so interoperability may be achieved.
1	Part 4, Section 7.4.2.5	-	te	This element defines values for use on Windows and Macintosh platforms, but not for Linux or any other operating system.	Several options here, but the desire is to allow cross platform interoperability.
1	Part 4, Section 7.4.2.5	-	te	Even within a single platform, there is not enough information given to achieve interoperability. For example, what are the allowed values and meanings for a “built-in Windows clipboard format value”?	Specify this so interoperability may be achieved.
1	Throughout	-	te	The name "Office Open XML" is often mistakenly called 'Open Office XML” implying a connection to the OpenOffice project which does not exist. This naming confusion has been documented and has occurred numerous times, including by analysts and even in Microsoft press releases and blogs. Since “Open Office” is the pre-existing name, by 6 years, Ecma should choose a new name, less apt to continue this confusion.	Change the name of Office Open XML to a name which is not confused with OpenOffice.

1	Throughout	-	te	From the overall document contents, it is acutely clear that no effort has been made in OOXML to start from the existing ISO standard for the representation of documents in XML, that is ODF 1.0, ISO/IEC 26300:2006. We can see no reason for that deliberate departure and contend that unneeded differences are harmful, and request that the OOXML proposal be rewritten starting from the existing standard.	Rewrite OOXML starting from ODF 1.0, ISO/IEC 26300:2006, for all matters that apply.