JTC-1 objections against ISO EOOXML

Please find objections to the ECMA 376 standard : http://www.ecma-international.org/publications/standards/Ecma-376.htm

The page numbers refer to the full **ECMA-376.pdf** document (6000+ pages) that was sent to all national bodies, and for which an archive can be found here :
http://web.mit.edu/~stevenj/www/ecma376.html

(As of February 8, the original ECMA-376.pdf document is not available on ecma-international.org anymore, replaced by 5 parts whose pagination has been changed).

Page 11 **Performance**	"Growth Hint A suggested number of bytes to reserve for the part to grow in-place. Optional. The package implementer might allow a growth hint to be provided by a producer. [O1.1]" objection ==> while optional, a consumer will have to use an appropriate ZIP library in order to maintain growth appropriately when updating a document in order to avoid hurting speed. Performance hit results from being forced to rebuild the package instead of making in-place edits. Most ZIP libraries out there don't allow a growth hint to be set.	
Page 16 **Cultural Adaptability problem**	"An Office Open XML part name shall contain only ASCII characters, in non-escaped or escaped form. The following ASCII characters are permitted in non-escaped form: "!", "$", "%", "&", "'", "(", ")", "*", "+", ",", "-", ".", the decimal digits "0"–"9", ":", ";", "=", "@", the Latin alphabetic characters "A"–"Z" and "a"–"z", "_", and "~". All other ASCII characters are permitted only when escaped as an encoded triplet of the form "%HH", where H is a hexadecimal digit.]" objection ==> unnecessary charset restriction, incompatible with the arbitrary nature of part names.	
Page 16 ***Poorly-designed packaging***	objection ==> lack of neutrality of Microsoft applications for any form of XML processing. If you create a file compatible with OOXML but whose part names do not follow the default part names created by Word/Excel/Powerpoint when a document is saved, then Word/Excel/Powerpoint reshuffle the entire underlying part structure so that it meets the default structure, thereby destroying the original one. Steps to reproduce the behavior : an empty Excel 2007 file is available here (http://xlsgen.arstdesign.com/special/ExcelWorkbook.xlsx). If you open this file in Excel 2007 and save it under a different name, you'll experience the part reshuffling. Just unzip both files, and notice how the structure has changed. Steps to produce the empty Excel 2007 file above : - start Excel 2007, create a new blank file, save it, close it - unzip the file - rename part "xl/worksheets/sheet1.xml" as "xl/custom/sheet1.xml" - edit the content of part "[Content_Types.xml]" and replace "/xl/worksheets/sheet1.xml" by "/xl/custom/sheet1.xml" in the body	

	- zip the file	
No Page number **Unspecified**	password-protection scheme of documents. objection ==> Password-protected documents are OLE documents, instead of ZIP files, and use proprietary encryption algorithms as well as an entirely undocumented encryption/decryption process. Note that password-protected an Excel document is different than password-protecting an Excel worksheet tab. In the latter case, the specs (page 2022) specifies the way to generate a hash key from a password.. A screenshot of a password-protection Excel 2007 document opened in an OLE document viewer is provided here : http://www.codeproject.com/cs/library/office2007bin/office2007bin9.gif	
Page 2022 **Under-specified**	"Specifies the hash of the password required for editing this worksheet. This protection is optional and may be ignored by applications that choose not to support this functionality." objection ==> does not specify the different types of password protection: password-protection of embedded VBA macros is not specified.	
No Page number **Unspecified**	embedded VBA macros are not mentioned. objection ==> how to read and write VBA macros from legacy documents. A screenshot of embedded VBA macros extracted and opened in an OLE document viewer is provided here : http://www.codeproject.com/cs/library/office2007bin/office2007bin5.gif	
Page 1897 **Under-specified**	"codeName (CodeName) Specifies the GUID that identifies the code project that is associated with the workbook. [*Note*: the primary use of this attribute is to track the version of the compiled code.] The possible values for this attribute are defined by the ST_Guid simple type (§3.18.41)." objection ==> the process to obtain and use the GUID associated to embedded VBA macros is not specified.	
Page 155 **Platform dependency**	"An instance of this part type contains information about the initialization and environment of a printer or a display device. The layout of this data structure is application-defined. [*Example*: An Office Open XML producer Shared on Windows might store the DEVMODE structure defined here: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gdi/prntspol_8nle.asp, while an application 3 on the Mac OS might choose to store the print record defined here: http://developer.apple.com/documentation/Carbon/Reference/CarbonPrinting Manager_Ref/Reference/refer ence.html. *end example*]" objection ==> platform-specific structure in printer settings that needs to be dealt with anyway by any consumer of an existing OOXML file. Printer settings are indeed integral part of the layout rendering mechanism on screen as well.	

Page 3588 **Under-specified**	"Specifies the length of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [Example: Consider a DrawingML object specified as follows:<… cx="1828800" cy="200000"/> The cx attributes specifies that this object has a height of 1828800 EMUs (English MetricUnits). end example] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§5.1.12.42). objection ==> the translation from the old undocumented, non-linear, MSO coordinate systems used for vector drawings in old Word/Excel/Powerpoint to the newer ones is not specified.	
Page 2084 **Performance**	"The calculation chain described in this section is not required by the spreadsheet application, but can be used if the spreadsheet application finds it useful. It can be loaded by a spreadsheet application, or the application may optionally construct it at run time in memory based on formula dependencies. Since the xml data described in this section is not strictly required, the spreadsheet application is free to ignore the order in which the calculation chain specifies calculations - i.e., even if the calculation chain is loaded, the spreadsheet application is free to perform calculations in a different order at run time." Objection ==> performance penalty for any writer that does not create the calculation chain. Without it, next time the document is opened in Excel, it will have to be entirely recalculated (which may not be doable if corresponding data sources are not available). Microsoft should document the process to derive the calculation chain from formulas.	
No Page number **Unspecified – Evidence of NO full-fidelity claim in practice**	chart drawing engine changes are unspecified, despite the fact that a chart created with an older version of Excel will display differently when opened in Excel 2007. Objection ==> chart drawing engine changes left unspecified. Example : a screenshot of this simple chart created with Excel 2003 (http://xlsgen.arstdesign.com/special/ChartExcel2003.jpg), then opened in Excel 2007 (http://xlsgen.arstdesign.com/special/ChartExcel2007.jpg). The example is available here (http://xlsgen.arstdesign.com/special/Chart97_BreakingChange2007.xls). Unexpected changes in this example are : - vertical axis all set to automatic scale/min/max. Also impacts the number of horizontal gridlines in the background. - chart title font not the same weight - chart title incorrectly positioned vertically. - legend border incorrect. - legend entries incorrectly positioned. - spacing between the plot area and the legend.	
No Page number **Unspecified**	Excel file round-trip scenarios are unspecified. If you create a fresh Excel 2007 file and add an object only available in the latest version, such as the in-place color data bar, and then save it as an Excel97-2003 compatible file, then Microsoft implementation creates new internal Excel 97-2003 compatible BIFF8 records to store the in-place color data bar, such that if the file is updated with an older version (where the in-place color data bar won't appear), and the file is eventually reopened in Excel 2007, then the in-place color data bar will still	

	show up. objection ==> all new BIFF8 records are left undocumented.	