# Open Document Format Alliance Response to the Burton Group's Report

The Burton Group has recently released a report entitled "What's Up, .DOC? ODF, OOXML, and the Revolutionary Implications of XML in Productivity Applications"[1], written by analysts Peter O'Kelly and Guy Creese. This report makes a number of erroneously negative and unbalanced statements about the Open Document Format (ODF), many of them quite puzzling to us.

The Burton Group chose not to review their report before publication with the ODF Alliance, the foremost industry organization, with 480+ members in 53 countries, supporting the Open Document Format for governments. Had they done so, we would have brought to their attention a number of factual errors in the report, pointed out what in our opinion were weaknesses in the logic, and noted places where they may have inadvertently been misled by their sources.

The analysis that follows provides, on a page-by-page basis, eighteen flaws with this report that we have identified so far. We raise, and clarify, these issues for the benefit of others, especially for the government officials with whom we work closely, who are evaluating their options to gain control over their documents and may be confused by some of this content.

1. Page 5 says *"...[L]ibraries and large businesses, faced with storing and using years of Microsoft Office legacy documents, will prefer OOXML, as OOXML can more faithfully recreate the look and metadata (such as spreadsheet formulas) stored in Microsoft's binary file formats."*

   This statement confuses file formats and applications. Surely, OOXML cannot faithfully recreate the look of anything. It is a file format, not an application.

   Microsoft Office is the application that interprets OOXML, and it can also render legacy binary file formats, except when Microsoft decides to remove support for them, as Microsoft recently did with Office 2003 SP3[2]. There can be further problems when Microsoft decides not to support legacy features, such as when they removed support for Visual Basic scripting in Office 2008 for the Macintosh[3].

   The authors fail to note that no other application supporting OOXML has been able to faithfully or fully recreate the look of Microsoft's legacy binary documents. So the statement that OOXML – a file format – is the solution for rendering legacy documents is simply false.

1 http://www.burtongroup.com/Guest/Ccs/WhatsUpDoc.aspx
2 http://www.eweek.com/c/a/Windows/Microsoft-Takes-Heat-for-Office-2003-SP3-File-Format-Blocking/
3 http://www.macworld.com/article/54320/2006/12/microsoft.html

2. Page 5 asserts that OOXML will win because, *"...OOXML is an extensible standard. It allows vendors and enterprises to extend the standard within an OOXML-defined framework... This built-in ability to augment the OOXML standard is a safety valve for future innovation, allowing new features to be added without forcing vendors to invent yet another separate file format or wait for standards bodies to give their approval."*

   Why does the report argue that OOXML is preferred over ODF for extensibility reasons, but fail to mention that ODF's extensibility features are just as rich? This lack of balance is disconcerting. ODF has extensibility features that are just as powerful as OOXML's, and, what is more important, ODF's features are based on existing industry standards, such as the W3C's RDF and XForms.

   Further, it is not clear to us what the statement about not waiting for standards body to give their approval means. Surely the authors are not suggesting that we repeat the problems of some earlier efforts where multiple "user defined extensions" to a standard meant that we really didn't have a standard at all?

3. Page 5 also discusses OOXML's support for "custom schemas" and suggests that "OOXML is more ecosystem- and application-oriented than ODF."

   We think it is unwise to consider exclusively Microsoft's particular "custom schema" feature. We should instead ask what the underlying business problem is that we are trying to solve.

   For example, if you wish to programmatically update a stock price in a document (what the paper suggests is solved by custom schemas) then there are many other approaches, from custom spreadsheet functions, to document transformation, to DOM manipulations, to tagging, to mapping to an XML database, and so on. The underlying problem is how to take what is ordinarily unstructured document content and impose an organization's structured view on it. Suggesting that only Microsoft's custom schema support solves the problem, while ignoring ODF's support for alternative and standards-based approaches to the same underlying business task, is misguided, in our opinion.

   In particular, the paper fails to note ODF's XForms support and how it can be used to enable users to add data to a document in a user-defined schema and then submit it to a web service. XForms is a W3C standard.[4] Here again, a sense of balance is missing, as is consideration of fully capable modern programming techniques that employ ODF to solve real business problems.

4. Page 5 says of the OOXML standard, in a section pointedly called *"What's In a Name?: Innuendo," that "IBM and Sun reference it via its Ecma name of "Office Open XML" (as reminder of its origins in the proprietary Microsoft Office file formats)."*

   This is a curious statement, but there is no need to suggest the name is used for some

4 http://www.w3.org/MarkUp/Forms/

propagandistic purpose. Many people call the standard “Office Open XML” because that is in fact its correct name, both in ISO, and in Ecma, and even its legal name as used by Microsoft in their own Open Specification Promise.[5]

Speaking of names, nothing is said about Microsoft choosing a name for their specification which was obviously destined to cause continued confusion with ODF's first implementation, Open Office.[6]

5. Page 12 says, *“...[M]any Microsoft competitors have attempted to exploit the transition to open, XML-based file formats in order to more effectively compete with Microsoft Office. While some of its competitors have very little hope of establishing successful products that directly compete with Microsoft Office, they may still seek to disrupt Microsoft's business by making Office less profitable, thus depriving Microsoft of opportunities to use Office profits to subsidize product development efforts in other market segments.”*

   The report seems to suggest that it is somehow improper to compete against Microsoft. We find that view puzzling and confusing. ODF vendors are not “exploiting” the transition to open, XML-based file formats. They were the source, the vanguard of the transition to open, XML-based file formats, having initiated standardization of ODF within OASIS over 5 years ago. Also, according to the authors' own statements earlier (bottom of page 6), OpenOffice has had 100 million downloads compared to Office's 500 million estimated users. How can it be said that there is “very little hope of establishing successful products that directly compete with Microsoft Office”?

   ODF is the source for new innovation and is set to further extend the value of modern technologies that have profoundly affected users of the Internet. For example, the Wikimedia Foundation, the technology arm of the organization that brought us Wikipedia, announced that they will use ODF to produce publication-quality documents from Wikipedia entries.[7]

6. Page 13 says, *“ODF is currently somewhat simple (and simplistic) compared with alternatives such as OOXML”*.

   The rather negative term “simplistic” is asserted with no backing argument or evident motivation. In our opinion, this does little to establish a sense of objectivity in the report. We might also ask when did “simple” became bad thing in a standard?

   XML is simple. HTML is simple.

   Moreover, Simple Mail Transfer Protocol (SMTP), Simple Network Management Protocol (SNMP) and other essential IETF standards are explicitly named that way because of their simplicity.

5 http://www.microsoft.com/interop/osp/default.mspx
6 http://www.robweir.com/blog/2007/01/amusing-but-still-confusing.html
7 http://wikimediafoundation.org/wiki/Wikis_Go_Printable

In our opinion, the best word to describe these global standards and ODF is not “simple,” but rather “elegant.”

There are considerable liabilities associated with unnecessary complexity: increased development and maintenance costs, increased difficulty to program, more complex (and therefore possibly negative) security implications, and generally buggier implementations. A more balanced approach would have examined the negative effects of a more complex standard.

“As simple as possible, but no simpler” was Einstein's advice, and is worth heeding. You can always build complex applications from simple pieces, but you cannot easily build simple applications from complex pieces.

7. Page 13 says, *“In terms of productivity application model concerns, ODF is primarily focused on content and presentation domains, and it is far less useful for scenarios requiring advanced structure and behavior capabilities. For example, ODF (currently in a 1.1 revision) supports a single table type for use within document, spreadsheet, and presentation applications”*

   The authors fail to make a convincing argument here, in our opinion. Taking a benefit of ODF and declaring that it is a liability, but without any argument, is puzzling. Most experts would say that, structurally, a table in a word processing document, a spreadsheet and a presentation share a number of factors in common that express their “tableness” such as an addressing mode by row and column, the ability to adjust row heights and column widths, the ability to have background colors, text content, etc.

   Good analysis and and elegant design, (an art not a science, but still one with established norms) calls for common abstractions to be commonly represented. So the fact that ODF has a single table model is a virtue of the format, not a liability.

   Note that there is nothing that requires the application to make a table in the user interface appear the same in all three applications. ODF is only defining the underlying storage model, and a table is a table is a table. There is no need for making this more complicated than necessary.

   Conversely, examining why a single standard unnecessarily includes redundant representations of the same basic concept in three different areas would be very useful, if it can be justified. It would lend balance to the arguments presented. Such redundancy would tend to make a standard much longer than expected and increase the development complexity, unless you already happen to have an implementation.

8. Further on page 13 the authors state, *“...it's essentially impossible to get ODF proposals approved if they're not also supported in OpenOffice.org, and further noted that Sun closely controls OpenOffice.org (much as it also holds control over Java).”*

   This is demonstrably false, and the use of unnamed “vendors” as sources does not eliminate the

need for doing basic fact checking on such claims. Rumors and innuendo do not objective analysis make.

First, on the control aspect, note that ODF 1.0, the standard, is owned and controlled by OASIS, a standards consortium of over 600 member organizations. Sun is just one company among many members. Indeed, for most of the development of ODF, Microsoft was on the Board of Directors of OASIS.

Second, OASIS is a corporation. It is legally bound to its Bylaws. There is no arbitrary control by member corporations.

The ODF TC is co-chaired by an IBM employee and a Sun employee, and is regulated by the OASIS TC Process document, which is publicly readable by all [8] and has clear rules of procedure and appeal.

The ODF TC has three subcommittees. The Accessibility SC is co-chaired by IBM and Sun, while the Formula Subcommittee and the Metadata Subcommittee are each chaired by individual members of OASIS who are not affiliated with any large corporations.

Voting rights in the ODF TC, for accepting or rejecting features, is currently as follows:

- Sun – 3 voting members
- IBM – 4 voting members
- Individuals – 3 voting members

This can easily be verified at the OASIS ODF TC website.[9]

Is sharing the chair position on the TC and on 1 of 3 subcommittees considered "closely controlling"? Is having 30% of the votes considered "closely controlling"?

As for proposals being accepted into ODF, we note that all three major features for ODF 1.2, RDF metadata, OpenFormula, and enhanced accessibility, are new proposals which have not been yet implemented in OpenOffice. Moreover, the ODF TC is currently processing a set of features requested by the KOffice open source project. So the assertion that it is "essentially impossible" to get new features into ODF if they are not already supported by OpenOffice is not true. This error is unfortunate and needs correcting through rigorous fact checking, as do the others, in our opinion.

Oddly enough, this particular error occurs in several places. A search of the report for the word "control" shows it used six times, once in reference to "Chinese communists" and five times in reference to Sun Microsystems.

Note, however, that no mention is ever made of the strong direct control Microsoft asserts over OOXML, its having sole chairmanship of the Ecma TC45, and its having secured a committee charter that prevents any changes to OOXML that are not compatible with Microsoft Office.

8 http://www.oasis-open.org/committees/process.php
9 http://www.oasis-open.org/committees/membership.php?wg_abbrev=office

Again, we're puzzled by the inaccuracy on one hand and the lack of balance on the other.

9. On page 14 the authors state, *"Note that it wasn't possible for Microsoft to embrace ODF during this period, as Microsoft's investment in XML file formats was long underway before ODF became an OASIS standard in 2005 (and because of Microsoft's need to maintain compatibility with earlier Microsoft file formats)."*

   This contradicts the author's own presented timeline. The ODF Technical Committee was created in OASIS back in 2002. There was clearly ample opportunity for Microsoft to join and influence the evolution of ODF. Since Microsoft was on the OASIS Board of Directors, they had significant opportunity to both know about and participate in what was happening in the development of ODF. It's unfortunate that they chose not to lend their expertise to this industry community effort.

10. Page 16, says *"Since ODF is less compatible with the earlier binary Microsoft file formats than OOXML, file format translations involving ODF can result in the loss of file fidelity, a constraint that limits ODF's utility for organizations that need to support file-based workflow involving Microsoft and non-Microsoft applications."*

    The statement that ODF cannot represent the contents of legacy documents with full fidelity is false. Further, it ascribes a fault to ODF as a format that is really an application issue.

    ODF, using the defined extensibility mechanisms in the ODF standard, can represent everything in Microsoft's legacy binary formats. What in practice makes this difficult is that the proprietary legacy document formats are poorly documented and have been historically withheld from competitors.

    On the other hand, OOXML itself cannot represent all legacy formats without extensions. Extensions are required for features such as scripts, macros, DRM, etc., as the authors point out earlier in the report in reference to .XLSM documents. But if ODF can use extensions, it is just as capable of representing legacy binary formats as OOXML is.

11. Page 17 says of the OpenDocument Foundation, that *"It was unable to successfully address requirements for capabilities related to workflows with Microsoft Office applications, however, and parted ways with the OASIS ODF-TC after it failed to gain approval for several proposals designed to make ODF more capable for addressing full-fidelity file format operations involving Office clients."*

    This statement is misguided in its omissions. First, the authors failed to note that the ODF TC evaluated the Foundation's proposals at great length, and were not persuaded that these changes would in fact bring the fidelity benefits claimed by the Foundation. Second, even some former Foundation members, when put to a vote, disapproved the proposal. Third, the vote was not against interoperability. It was an evaluation by the technical experts on the TC that the proposal would not work.

This was clearly a controversial issue. Why was only one side presented? Where was the balance? Where was the confirmation from multiple independent experts?

It would have been interesting for the report to give a historical examination of compatibility of Microsoft's legacy document formats between the various implementations of Microsoft's own software. Have users always had full-fidelity compatibility between these releases? Also of interest might be an analysis of what happened to users who decided to employ Microsoft's Office 2003 XML format, now deprecated.

12. On page 17, the report says, *"Despite all of the debate and controversy surrounding ODF and OOXML, it's important to recognize that the standards organizations are working as designed, and that both the standards and the organizations are constructively evolving as a result."* And later on that same page, *"It's likely ISO will revise its procedures to prevent similar disruptions in the future, so in some respects the OOXML episode will produce some useful stimulus/response improvements within ISO."*

    This statement misses the point. The question is not whether the organizations' procedures are perfect or not. No matter how well intentioned a process is, loopholes can always be found. Sometimes they are small, sometimes they are huge, but in all cases exploiting those loopholes is nothing less than contradicting the spirit of the process. The authors fail to consider the numerous articles highlighting how those loopholes have been exploited, and the spirit of the process subverted by stacking committees, by offering marketing consideration to business partner members who voted in favor of OOXML, and by overloading fast track processes with proposals that are of extraordinary and unsuitable length.

    The statement that this is all useful and constructive because it will be a stimulus to improvement within ISO is patronizing of ISO. It ignores the basic issue: wouldn't it have been better to have avoided the extreme exploitation of process loopholes and followed the path of previous interpretations of the spirit of the processes as others have done in the past?

    Also see the article "Sweden's SIS Declares OOXML Vote Invalid - Will Change Vote from Yes to Abstain – Updated."[10]

13. At the bottom of page 17, the authors present a list of prior protocol and API disputes with Microsoft, such as VIM, IDAPI, OpenDoc, etc., and suggest, *"In these and other earlier encounters, when Microsoft's competitors sought to collectively compete with Microsoft by leveraging standards, the everybody-but-Microsoft standards have generally failed to achieve significant market momentum."*

    The examples given are ancient history. Also, note that VIM (1993), IDAPI (1992) and OpenDoc (1992) were not formal standards. The report fails to note that the market has evolved in the past 15 years and that that customers are better educated in the liabilities of vendor lock-in, and are now more aware of the important role that open standards play in ensuring

10 http://www.groklaw.net/articlebasic.php?story=20070830155109351

interoperability. Whereas 15 years ago, it was enough to require the use of COTS software, today procurement is increasingly requiring the use of open standards.

In our opinion, it would be far better to look at the example of Internet and web standards to see how Microsoft's competitors, and Microsoft itself, have succeeded. ODF has the potential to be the same competitive force for office productivity applications, using the same standards-based approach.

14. On page 18 the authors introduce a section by saying, *"If productivity application evolution had peaked around the feature set of Office 97, and if there weren't exabytes of files captured in earlier Microsoft Office file formats, it's possible that ODF could have succeeded as a global standard for productivity application file formats."*

    This point is not argued, it is just asserted, without any supporting evidence. We suspect that many of the arguments being made today in the report for Microsoft and against ODF would also have been made in 1997 had we been in that situation then. The network effects were smaller, but they were big for the time.

15. Later on page 18 the authors state, *"OOXML will be more pervasive than ODF for several reasons. It's a better form-follows-function fit for most productivity application usage patterns"*.

    This is a bold claim without argument or analysis to back it up, as far as we understand it. On what basis is the assertion that more than 50% ("most") uses of documents cannot be represented in ODF? We believe, for example, that the entire Burton Group report could be represented in ODF without any loss in fidelity. Do the authors believe that more than 50% of the documents in use, today and legacy, are more complex than their 37 page report?

16. Further on page 18, the authors note, *"OOXML will gain market momentum as vendors such as Altova and Mindjet introduce products that support OOXML. Altova, the leading vendor of tools for XML-focused developers and designers, added OOXML to its XMLSpy product line in 2007."*

    This statement misrepresents market momentum by citing niche application support. While they may be fine products, will it really be the case that an XML programmer's editor and mind map software will drive the adoption of an XML office document format like OOXML?

    Will someone in a university or a business office adopt OOXML as a word processor format because they can edit it in Altova? That would be remarkable, considering the relatively small percentage of all end users who edit XML at that level. MindJet already supports the legacy binary Office formats. By what mechanism does MindJet supporting OOXML cause someone to move to OOXML?

    Certainly, if a user wants to move to OOXML, having this support removes a potential obstacle to migration. But it alone can do nothing to cause market momentum. We believe, in fact, that

many users will simply stay with the .doc, .xls, and .ppt formats if they wish to use Microsoft formats, as was recommended by the recent Becta report in the UK.[11]

17. Page 19 says that *"ODF represents laudable design and standards work. It's a clean and useful design, but it's appropriate mostly for relatively unusual scenarios in which full Microsoft Office file format fidelity isn't a requirement."*

    The authors here contradict their own argument. On page 8 they stated, "In many cases, the productivity application file is ephemeral, used only to present and capture user actions for business transactions that are ultimately captured in enterprise systems rather than stand-alone productivity application files (that is, the files are destroyed when the transaction is complete and captured in other systems of record)." We believe, and follow the practice of, authoring a document in ODF and publishing it as a PDF file. For us this is a not an "unusual scenario" nor does it necessarily involve "full Microsoft Office file format fidelity,"

    Further, what does "full Microsoft Office file format fidelity" even mean? This seems like a circular argument, that ODF is only useful in cases where you are not using Microsoft Office file formats.

18. On page 21, the authors claim, "*Even Adobe's own Buzzword service will likely add support for OOXML—again, if only for Office 2007 file compatibility.*"

    This tells only half the story, again. Adobe has stated publicly that it plans on adding ODF support.[12] The cherry-picking of statements, ignoring positive facts about ODF, while speculating on positive futures for OOXML, is puzzling and goes against the paper's claimed position of objectivity and vendor neutrality.

There are also issues not raised that cause concerns. For instance, why is there no discussion on accessibility issues raised by the disability community with respect to office documents? These issues have raised worldwide consciousness of the impact of information technology decisions and standards on the lives of people with disabilities. We hope it is not because the authors did not wish to acknowledge either (1) the issue's importance, or (2) that ODF v1.1 has established a new high water mark for document formats, a high water mark that should not be allowed to recede with the acceptance of anything less from any other office document format.[13]

We believe every statement in this report should be re-examined to ensure that it is fully balanced and has taken in the views of all relevant stakeholders. In this quick review, we believe we have shown that the analysis is incomplete, at times misguided, and selective. Please review this paper critically along with "Achieving Openness: A closer look at ODF and OOXML"[14] and come to your own conclusions.

11 http://publications.becta.org.uk/download.cfm?resID=35275
12 http://www.adobe.com/special/buzzword/faq.html
13 Http://atrc.utoronto.ca/index.php?option=com_contect&sectionid=14&task=view&hidemainmenu=1&id=371
14 http://www.onlamp.com/pub/a/onlamp/2007/06/14/achieving-openness-a-closer-look-at-odf-and-ooxml.html?page=1