

UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <http://www.upgrade-cepis.org/>

Publisher

UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>) by **Novática** (<http://www.ati.es/novatica/>), journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*, <http://www.ati.es/>)

UPGRADE monographs are also published in Spanish (full version printed; summary, abstracts and some articles online) by **Novática**

UPGRADE was created in October 2000 by CEPIS and was first published by **Novática** and **INFORMATIK/INFORMATIQUE**, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <http://www.svifi.ch/>)

UPGRADE is the anchor point for UPENET (UPGRADE European NETWORK), the network of CEPIS member societies' publications, that currently includes the following ones:

- **Informatik-Spektrum**, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
- **ITNOW**, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
- **Mondo Digitale**, digital journal from the Italian CEPIS society AICA
- **Novática**, journal from the Spanish CEPIS society ATI
- **OCG Journal**, journal from the Austrian CEPIS society OCG
- **Pliroforiki**, journal from the Cyprus CEPIS society CCS
- **Pro Dialog**, journal from the Polish CEPIS society PTI-PIPS

Editorial Team

Chief Editor: Llorenç Pagés-Casas, Spain, pages@ati.es

Associate Editors:

François Louis Nicolet, Switzerland, nicolet@acm.org
Roberto Carniel, Italy, carniel@dgf.uniud.it
Zakaria Maamar, Arab Emirates, Zakaria.Maamar@zu.ac.ae
Soraya Kouadri Mostéfaoui, Switzerland, soraya.kouadrimostefaoui@gmail.com
Rafael Fernández Calvo, Spain, rfcvalvo@ati.es

Editorial Board

Prof. Wolfried Stucky, CEPIS Former President
Prof. Nello Scarabottolo, CEPIS Vice President
Fernando Piera Gómez and Llorenç Pagés-Casas, ATI (Spain)
François Louis Nicolet, SI (Switzerland)
Roberto Carniel, ALSI - Tecnoteca (Italy)

UPENET Advisory Board

Hermann Engesser (Informatik-Spektrum, Germany and Switzerland)
Brian Runciman (ITNOW, United Kingdom)
Franco Filippazzi (Mondo Digitale, Italy)
Llorenç Pagés-Casas (Novática, Spain)
Veith Risak (OCG Journal, Austria)
Panicos Masouras (Pliroforiki, Cyprus)
Andrzej Marciniak (Pro Dialog, Poland)
Rafael Fernández Calvo (Coordination)

English Language Editors: Mike Andersson, Richard Butchart, David Cash, Arthur Cook, Tracey Darch, Laura Davies, Nick Dunn, Rodney Fennemore, Hilary Green, Roger Harris, Michael Hird, Jim Holder, Alasdair MacLeod, Pat Moody, Adam David Moss, Phil Parkin, Brian Robson

Cover page designed by RFCalvo, © Rafael Fernández Calvo 2006

Layout Design: François Louis Nicolet

Composition: Jorge Llácer-Gil de Ramales

Editorial correspondence: Llorenç.Pagés-Casas@ati.es

Advertising correspondence: novatica@ati.es

UPGRADE Newslist available at

<http://www.upgrade-cepis.org/pages/editinfo.html#newslist>

Copyright

© Novática 2006 (for the monograph)

© CEPIS 2006 (for the sections UPENET and CEPIS News)

All rights reserved under otherwise stated. Abstracting is permitted with credit to the source. For copying, reprint, or republication permission, contact the Editorial Team

The opinions expressed by the authors are their exclusive responsibility

ISSN 1684-5285

Monograph of next issue (February 2007)

"Next Generation Web Search"

(The full schedule of UPGRADE is available at our website)



The European Journal for the Informatics Professional
<http://www.upgrade-cepis.org>

Vol. VII, issue No. 6, December 2006

- 2 Editorial: UPGRADE in Top Position of Google PageRank for ICT Journals — *Geoff McMullen*

Monograph: Open Document Format (ODF) (published jointly with Novática*)

Guest Editors: *Jesús Tramullas-Saz, Piedad Garrido-Picazo, Marco Fioretti*

- 3 Presentation: OpenDocument Standard for Digital Documents — *Jesús Tramullas-Saz and Piedad Garrido-Picazo*
- 5 Open by Design: The OpenDocument Format Standard for Office Applications — *Erwin Tenhumberg, Donald Harbison, and Rob Weir*
- 14 Is OpenDocument an Open Standard? Yes! — *David A. Wheeler*
- 25 OpenDocument Hidden Traps and their Side Effects on Free/Open Source Software — *Marco Fioretti*
- 29 ISO-26300 (OpenDocument) vs. MS-Office Open XML — *Alberto Barrionuevo-García*
- 38 Interoperability: Will the Real Universal File Format please Stand Up? — *Sam Hiser and Gary Edwards*
- 47 ODF: The Emerging Document Format of Choice for Governments — *Marino Marcich*
- 50 Promotion of the Use of Open Document Formats by the IDA and IDABC Programmes — *Miguel A. Amutio-Gómez*
- 53 A Brief History of Open Standards in Denmark — *John Gøtze*
- 57 Standard Open Formats and Libre Software in the Extremadura Public Administration — *Luis Millán-Vázquez de Miguel*

UPENET (UPGRADE European NETWORK)

- 60 From **Mondo Digitale** (From AICA, Italy)
Programming Languages
Programming Languages: An Introduction — *Carlo Ghezzi*
- 64 From **Pro Dialog** (PTI-PIPS, Poland)
E-Commerce
Organization and Economics of Entertainment Services Networks Exchanging Virtual Goods — *Andrzej P. Urbański*

CEPIS NEWS

- 70 Harmonise Project: Assigning Responsibilities among the Partners and Planning Next Steps — *François-Philippe Draguet*
- 71 News & Events: European Funded Projects and News Updates

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>.

ISO-26300 (OpenDocument) vs. MS-Office Open XML

Alberto Barrionuevo-García

Until recently, Humankind has been using electronic document formats that were exclusive and which locked users into a specific IT provider and their software. These providers have been changing over the years, with the exception of the last ten years. These changes have resulted in millions of unusable and inaccessible documents due to formats that are obsolete or not properly readable by the new applications. These days, and for the last ten years or more, MS-Office has been the dominant operator, tying users to an operating system from the same manufacturer and at the same time shutting out most other operators. However, the arrival of open standard ISO 26300, OpenDocument, looks like changing this outlook and releasing application formats and their users from the stranglehold of a small number of applications and systems. In response to this radical sea change in the market, the company that up until now has dominated the office automation market is proposing an alternative, semi-open format. Much of the future of information technology and the path that the Information Age will take now depends on which format gets the upper hand in this fascinating battle, the like of which has never been seen in the history of computing. In this article the two formats are compared from a number of viewpoints and one basic conclusion is reached: only one should survive.



Opentia, S.L., 2007. This article is distributed under the "Creative Commons Attribution-ShareAlike-NoDerivs-NoComercial 2.5 Spain License" Creative Commons licence, available at <<http://creativecommons.org/licenses/by-nc-nd/2.5/deed.en>>.

Keywords: ECMA, ECMA-Office Open XML, Electronic Document, ISO/IEC, Libre Software, MS-Office Open XML, ODF, Office Automation Format, OpenDocument, Open Standards, Standard.

1. Historical Context

1.1. The Middle Ages of Electronic Documentation

From the moment when we stopped using "vi", "emacs", and "edit" to draft documents in ASCII with no more layout than could be achieved in a rudimentary and artistic manner by inserting or removing spaces and by using some special characters as decoration, office automation has been dominated by document formats with four main characteristics:

1) The format was dependent on the application that generated it and, therefore, on the company that owned that application.

2) The format was secret and exclusive to each particular manufacturer and, at most, other competitors or developers managed to support it by using costly reverse engineering techniques or by handing over large sums of money and signing agreements not to reveal the secrets of the format.

3) In practice, "in order to be compatible", everyone was obliged to use the format (and the application) of whichever was the dominant operator in the market at that time.

4) Generally speaking, these applications were sold under non-libre licences, and their code was as much a secret as the format itself.

Since the eighties, barring a few exceptional cases such as LaTeX, DocBook (for technical applications) and, much later, the OpenOffice formats, all users and buyers of office

Author

Alberto Barrionuevo-García is a graduate in Computer Science from the *Universidad Politécnica de Madrid*. He is a partner and CEO of OPENTIA, S.L. specialized in consultancy services related to computer standards, document management and archival, and open solutions in general, and also advanced migrations libre software through its subsidiary VIRTUA, S.L. He is Vice-President of the Munich-based European association, Federation for Free Information Infrastructures (FFII), and is the coordinator of the association's "Open Standards" working group. He is also working group and web coordinator of EstandaresAbiertos.org, member of the OpenDocument Fellowship, corporate member of the ODF Alliance, and member of Hispalinux. He is "Linux Registered User #130136". He has been director of operations and consulting at OpenText Corporation for Latin America; managing director, director of operations, and director of consulting at IXOS Software AG for Latin America; and technical consultant at SAP Spain and Portugal. <abarrio@ffii.org>.

automation software have effectively been forced always to use whichever application was dominating the market at that time so as to be able to exchange documents with the rest of society. This happened with applications such as WordStar first of all, followed by WordPerfect, and then, to a lesser extent, AmiPro, and finally the software which ended up dominating the market thanks to the virtual monopoly of its manufacturer in PC operating systems: MS-Office.

The feature common to all these formats is that if someone ever sent you a document in a format and/or version other than the application that you were using, you were forced to go out and get hold of the application behind that

format to be able to access the information it contained.

In the home computer market the truth is that getting access to new software would not, in all probability, be much of a problem because you could always trot along to your local purveyor of illegal software who would supply you with whatever you needed very cheaply or even for free. However, in the institutional and business market, the acceptance of a new document format has always meant taking a fairly important step, involving a mass migration of applications at every workstation, not to mention the consequent disbursement of the by no means insignificant cost of licences, training, and even the adaptation of existing documents and applications.

By way of a curious aside, at this point I always remember the notary I usually work with who "still" uses the non-graphical version of WordPerfect. They probably don't have to exchange much documentation with the outside world (I know for a fact that they don't use e-mail). But these are customs from bygone days and should have no place in our post-industrial era.

1.2 The Information Revolution: the Internet and New Opportunities

The obscurantist model of binding the application to the format persisted, without any real alternative, until the general population started getting hooked up to the Internet en masse, and formats that were application-independent, and which in many cases were not subject to any royalties, began to become popular.

The reason for this popularity was that their original creators knew that it would be impossible to compete with the dominant application (by now a virtual monopoly) unless they adopted the tactic of breaking the link between the format and the application, and released the format specification so that other applications could also use it, thereby benefiting from a greater economy of scale.

The most paradigmatic case is perhaps the situation of Sun in 1999, when it employed 40,000 people each with their corresponding workstation. When acquiring an office automation application, Sun was faced with the choice of either paying Microsoft for 40,000 licences, while also having to duplicate hardware for its many employees who needed to continue using Unix, or doing what they actually did: they bought the code of an office automation application on the German market, StarDivision, which also provided them with a marketable alternative to Microsoft's dominant product.

But a problem arose: an application like StarOffice was costly to maintain and keep up to date and would require a large department of developers devoted exclusively to it if they wanted to compete at the same level as other rival applications who did have such a department.

So, as a solution to that problem came the second major milestone of the short history of electronic document formats. It happened in the late nineties, just as a new model of development and distribution of desktop software was reaching maturity: libre software. Until a short time previ-

ously, libre software had been focusing its efforts on very technical applications, mainly aimed at servers and developers. But the time had come for libre software to take the plunge into the world of the desktop.

Thus, just one year after buying StarOffice, Sun took the eminently financial and commercial decision to release not only the specifications of the StarOffice format, but also the code used to implement those specifications in order to create a parallel libre software project that would maintain, use, and disseminate their formats. It was the launch of OpenOffice.org project, the same format as I am using to write this article.

The fact of the matter is that it was the only economically viable way that a hardware company such as Sun could compete against the practically monopolistic software that had all users locked in tight: i.e. Microsoft's.

At the same time, Sun decided to go on marketing their closed-source version of StarOffice based on OpenOffice.org's libre code with some extra functionalities. This dual licence and product model had already been used by another application that had been driven out of the market by Microsoft's stranglehold on operating systems: the forgotten Netscape, the seed from which, phoenix-like, the current successful Mozilla-Firefox web browser arose.

All this finally led to the commercial availability of a number of open document formats covering all the functionalities of the dominant application which could also be used by many of the other alternative office automation applications that already existed on the market..., as indeed has happened.

2 Document Standards

2.1 And why not an Open Standard?: Open Document

After a few years of putting up with some much vaunted computing standards that were too theoretical, complex, and flawed to work with, the official international standardization organizations decided to create some standards based on market-driven protocols or formats which had already proven their worth in the real world. This was probably helped by the fact that many companies were getting together to create their own alternative, non-governmental organizations for developing industrial standards. This was the case, for example, of the organization OASIS (*Organization for the Advancement of Structured Information Standards*), made up of almost all the major software developers around the world.

And this has also been the case of the document format created and maintained by OASIS, one which is independent, open and intended for use by anyone, and is based on the old StarOffice and OpenOffice.org formats: the OpenDocument format.

OASIS itself describes the purpose of the OpenDocument Committee¹ since its inception in late 2002 as "*to create an open, XML-based file format specification for office applications*". In other words, the format should be generic and aimed at all office automation applications

which, of course, voluntarily decide to use it. Later in this article we will see that today this is practically all of them, with one notable exception.

But in any event, what is really important and what makes a difference is that the standard was not created in the of the format and interests of any one application or manufacturer, but was corrected and reshaped where necessary in a manner that was totally open to all interested parties. And the result is that now, just over a year later, it is already being used natively² by at least 5 office suites covering the 5 most commonly used platforms (in descending order: Symbian, Windows, MacOS, Linux and FreeBSD), as well as several dozen applications more that use some of its parts (text document, presentation, spreadsheet, and vectorial drawing) or use it for some additional functionality (document management, indexing, forms, operating flows, etc.)

One factor that has definitely contributed to the ease with which many existing office automation applications have been able to adopt OpenDocument is that, as far as possible, its specification used any existing standards that were of interest. Also, for many of those standards there was a great deal of free source code and libraries available on the market. This made it possible to make implementations of the specification cheaper and easier. Examples of reused standards in OpenDocument are XML, ZIP, Dublin Core, SVG, XSL, SMIL, XLink, XForms, MathML, HTML, etc. This clearly shows there was no unnecessary re-inventing of the wheel. And testimony to this is the fact that the specification document³ is just a tad over 600 pages long

But aside from its many technical virtues which we will not go into right now, much of the interest that the format has aroused is down to its ratification as a (*de iure*) international standard by the ISO/IEC (*International Organization for Standardization / International Electrotechnical Commission*) on May 1, 2006. Since then, despite not being officially published until December 2006, OpenDocument has been approved as official standard ISO 26300.

After receiving such a unanimous "blessing", and in a period of a little over six months, a not inconsiderable number of governments and official bodies have decided to

adopt OpenDocument as their sole document standard. This is only logical, as there is no other standard of similar characteristics to serve as an alternative. In fact, it could even be argued that society has no need for there to be an alternative, in the same way that, for example, society has no need whatsoever for there to be an alternative to the metric measurement system. Here, a single standard fulfils its purpose perfectly; measurements are standardized and can be exchanged between entities and people without creating any unnecessary confusion due to the existence of a duality of systems. And if you need any proof of that, we refer you to the huge bungle that caused NASA's *Mars Climate* to crash⁴.

Among the most noteworthy cases of OpenDocument adoption are:

- The Massachusetts state government, perhaps the most conflictive and publicized case due to the great political and media pressure brought to bear by Microsoft in an attempt to overturn the decision.

- The Belgians who adopted the OpenDocument format throughout their federal government.

- Denmark's open standards law, on the back of the Ramboll report⁵ on the savings to be made by adopting OpenDocument as a standard format.

- The Motion of the Extremadura Regional Government in Spain mandating the exclusive use of OpenDocument and PDF/A in its government offices.

- The announcement by the French government that it is to migrate all its administration to OpenOffice.org software supporting OpenDocument, in the light of the conclusions of the Carayon Report⁶.

- Similar announcements by Norway's Ministry of Science.

- Several mass migrations in Brazilian government agencies.

- The mass migration of the Government of Tamil Nadu in India.

- The adoption of OpenDocument by the Zaragoza City Council in Spain.

- The exclusive adoption by the Misiones Province of Argentina.

- The pioneering legislation of the *Universidad de Cádiz* (Spain) governing electronic document exchange.

- Etc.

¹ OASIS's OpenDocument committee, created in November 2002. <<http://www.oasis-open.org/committees/office/charter.php>> (last accessed 29-12-2006).

² Software applications supporting OpenDocument or use it natively, according to the OpenDocument Fellowship. <<http://www.opendocumentfellowship.org/applications>> (last accessed 29-12-2006).

³ OpenDocument specification by OASIS. <<http://www.oasis-open.org/committees/download.php/19275/OpenDocument-v1.0ed2-cs1.odt>> (last accessed 29-12-2006).

⁴ CNN: "Metric mishap caused loss of NASA orbiter." <<http://www.oasis-open.org/committees/download.php/19275/OpenDocument-v1.0ed2-cs1.odt>> (last accessed 01-02-2007).

⁵ Danish Government: "Ramboll Report" on the cost savings from implementing OpenDocument instead of MS-Office Open XML. <http://www.osl.dk/upload-mappe/ram_engPDF/> (last accessed 29-12-2006).

⁶ French Government: "Carayon Report" on the cost savings from implementing OpenDocument. <<http://www.ladocumentationfrancaise.fr/rapports-publics/064000728/index.shtml>> (last accessed 10-01-2007).

The *OpenDocument Fellowship*⁷ and the *OpenOffice.org* project⁸ are monitoring these and other large scale adoptions of the standard, especially in the public sector. It is in the public sector where it is most necessary to use open standards in order to avoid discriminating among citizens on the basis of whether or not they are customers of a particular private provider by using a format that is exclusive to their own commercial application.

2.2 Do we Need a Second Standard? Microsoft's Strategy

When taking any decision it is normally impossible to satisfy all interests. But it is especially hard, and not entirely fair, to satisfy economic interests that are not only in private hands but, what is worse, are also contrary to the interests of society at large.

The company that holds a practical monopoly on desktop operating systems and office automation applications with its well-known duo Windows/Office, has decided not to adopt the OpenDocument standard adopted by practically all the other players in the market, and is now engaged in a process of intense lobbying to standardize their alternative format: Microsoft-Office Open XML.

As part of their strategy, Microsoft has used a European standardization agency with a reputation for being a soft touch: the ECMA (*European Computer Manufacturers Association*). Thus by means of a small technical committee, chaired by MS, in which most of the members were appointed expressly and solely for the purpose of this process⁹, they have pushed through the approval of the format under the name of *ECMA-Office Open XML* (EooX), though not without a resounding vote against from IBM, a historical member of the association. Now the next step is to push for its approval as an international standard by ISO, since ECMA is ultimately no more than an association of companies issuing their own private standards with no official status. However, many important voices have been raised against the idea of unnecessarily duplicating formats by having two standards¹⁰ that are completely overlapping and meet an identical need. Neither should the predictions of leading analysts¹¹ in this regard be ignored.

But the most interesting thing as far as this article is concerned is that the *ECMA-Office Open XML* standardiza-

tion effort carried out by ECMA has revealed a number of details that we need to bear in mind and which require studying in some detail.

Firstly, we should take a close look at the composition of ECMA's Technical Committee 45 convened to standardize Microsoft's format. As we have already mentioned, only four of its members are from ECMA (Microsoft, Intel, Toshiba and Apple) and the rest are external committee members expressly and solely appointed to the committee for the purpose of this standardization process.

But what is even more significant and revealing is the officially declared goal of the committee convened to standardize *ECMA-Office Open XML* format. It especially contrasts with the fair and open purpose declared by the OASIS committee that created OpenDocument, quoted earlier. The declared goal of the ECMA committee is none other than "*to produce a formal standard for office productivity applications within the Ecma International standards process which is fully compatible with the Office Open XML Formats.*"

The first thing that springs to mind when reading this strange declaration of purpose is that ECMA should at least have left out the adverb "fully"; that way we might have entertained the slightest of doubts as to the impartiality and neutrality of the committee's purpose. Thus, in the light of these declared intentions, the obvious question is: *in that case, why were the other entities on the committee at all, if it was merely a question of ratifying, without discussion, everything that Microsoft put on the table arising out of that "full compatibility" with the application belonging to the monopolistic provider?* For example, the case of Novell, one of the non-ECMA members of the committee, is significant. After a multimillion dollar patent agreement with Microsoft, perfectly interpretable as a "financing agreement", Novell designated themselves as developers of the support to the EooX format for OpenOffice.org. The idea is probably to avoid the impression that only one office automation application on the market will actually support it: the new, about to be released (and bold¹²) version of MS-Office.

However, Novell will have no easy task since, as opposed to the scarcely 700 pages of the OpenDocument specification, the final specification of the format proposed by

⁷ OpenDocument Fellowship: examples of the adoption of OpenDocument around the world. <<http://www.opendocumentfellowship.org/government/precedent>> (last accessed 29-12-2006).

⁸ OpenOffice.org: market share figures. <http://wiki.services.openoffice.org/wiki/Market_Share_Analysis> (last accessed 29-12-2006).

⁹ ECMA: Presentation "ECMA TC45 – Office Open XML Formats" (p 16). <http://www.ecma-international.org/activities/Office%20Open%20XML%20Formats/TC45_GA_Dez05.pdf> (last accessed 10-01-2007).

¹⁰ European Commission, "PEGSO Conclusions and Recommendations on Open Document Exchange Formats", 6-12-2006. <<http://europa.eu.int/idabc/in/document/3439>>.

¹¹ Report in which the Gartner Group predicts that the Microsoft format will not be standardized by ISO. <http://www.gartner.com/resources/140100/140101/iso_approval_of_oasis_opendo_140101.pdf> (last accessed 29-12-2006).

¹² Wall Street Journal, "Bold redesign improves Microsoft Office 2007", in which Microsoft is referred to as "bold" due to the high training costs that the major changes to the new MS-Office's user interface will involve. <http://www.moneyweb.co.za/shares/international_news/553820.htm> (last accessed 10-01-2007).

Microsoft and released by ECMA runs to over 6,000 pages to which will have to be added a welter of support documentation, all to cover no more and no less than the same functionality as the ODF.

Although the fact of the matter is that they do not work as well as each other, whatever well publicized claims may have originally been made. The fact is that one of them works significantly less well as a document format than the other. This may be one of the reasons why its specification is so inordinately hefty.

3. OpenDocument vs. ECMA-Office Open XML (EOOX)

When comparing the two formats the first thing that stands out is the different purposes of the standardization committees that released them. As we have mentioned, while one format is aimed at the industry as a whole with no reference or preference to any particular party, the other is aimed exclusively at being compatible with a specific application which is the exclusive property of the company that created the format. And, what is more, the sole purpose of that format is, in all probability, to keep the present users of their applications locked in to their products.

In fact it has gone to such lengths to ensure compatibility with MS-Office that EOOX's specification includes historical errors that are known by all, and makes no attempt to correct them¹³.

This is probably one of the few times in history that a standardization body has not removed known bugs from a format or protocol during the standardization process but has instead perpetuated them so that all other implementers of the specification will also be forced to artificially reproduce them in their respective applications. This leads us to the typical situation that arises when a dominant company has a problem. It has two ways of preventing negative repercussions, which is the real issue, not the problem in itself. The first option would be to simply solve the problem, while the other option is to export the problem to all the other competitors so that no one gains an advantage, even though this means that the users suffer in the process.

Another thing is that, to a large extent, the EOOX specification reinvents the wheel by avoiding the reuse of existing standards to cover included functionalities. This is probably for reasons of self-interest, as it is a way to inflict an additional effort and investment on the competition, while Microsoft requires no extra effort whatsoever as they already have the application and the fact that it works in such an unorthodox manner does not affect them. A good example is the case of hyperlinks. Rather than use the very widely and commonly used XLink standard, it encodes them in this peculiar and cryptic manner:

```
<w:hyperlink w:rel="rId1" w:history="1">
  <w:r>
    <w:t>This is a hyperlink</w:t>
  </w:r>
</w:hyperlink>
```

It is necessary to fetch the hyperlink in question from an additional file.

OpenDocument on the other hand is designed with human beings in mind and uses the XLink standard. Let us compare the previous code with the same hyperlink as encoded in OpenDocument:

```
<text:a xlink:href="http://example.com">
  This is a hyperlink
</text:a>
```

An example of the reuse of anything already existing and useful is that for any programmer familiar with XHTML, DocBook or even HTML, OpenDocument is quite similar and easy to implement. EOOX, on the other hand, encodes everything in a completely different and cryptic way, a way that any professional would find very difficult to understand and, what is worse, one that is practically impossible to translate into any other standard format such as XHTML or OpenDocument.

Hence the relatively well-founded suspicion that EOOX is simply a translation to XML of the secret binary formats that Microsoft have been using up to now, which are ultimately no more than simple and extremely technified dumps of the representation of the document to the memory of the MS-Office application; i.e. machine readable, but not human friendly. Something akin to programming a large software application in Assembler.

This view gains weight if we look at the different XML encoding methods used in the two formats. While OpenDocument uses a "mixed content" XML of the type used for non-structured content (as text documents are), EOOX organizes the inside of its files using an "unmixed content" method more typical of databases and structured data formats. To be more precise, in an XML document you can find two kinds of inputs, tags and text, and in between the opening tag and the closing tag, other tags and text can be included. Thus, in the mixed content method, in between the opening tag and the closing tag you can add either tags or text or a mixture of both. However, in the unmixed content method, you can only add more tags or more text, but not a mixture of both.

Let's see how this affects the representation of the same short text in OpenDocument and in EOOX. First of all, here is the original text with the format we wish to represent:

¹³ MS-Excel contains an error in the sequential calculation of the days of the calendar because it is based on the false premise that the year 1900, the origin year, was a leap year. In ECMA-Office Open XML this error is perpetuated. More information in Rob Weir's "A Leap Back". <<http://www.robweir.com/blog/2006/10/leap-back.html>> (last accessed 29-12-2006).

Esto es un documento muy simple con un poco de formato y un hiperenlace.¹⁴

And the hyperlink, which we cannot see in the example, is pointing to a URL `<http://www.estandaresabiertos.com/>`.

Thus, OpenDocument would represent this text in this way:

```
<text:p text:style-name="Standard">
Esto es un
<text:span text:style-name="T1">documento</
text:span>
muy simple
<text:span text:style-name="T2">con un poco
de</text:span>
formato y un
<text:a xlink:href="http://www.
EstandaresAbiertos.org">hiperenlace</
text:a>
</text:p>
```

which, as we can see, could even be modified manually using a text editor on any text terminal, as it is reasonably intelligible for anyone remotely familiar with HTML, XHTML, or Docbook.

Now let's see what EOOX encoding makes of the same text:

```
<w:p>
  <w:r>
    <w:t>Esto es un </w:t>
  </w:r>
  <w:r>
    <w:rPr>
      <w:b />
    </w:rPr>
    <w:t>documento</w:t>
  </w:r>
  <w:r>
    <w:t> muy simple </w:t>
  </w:r>
  <w:r>
    <w:rPr>
      <w:i />
    </w:rPr>
    <w:t>con un poco de</w:t>
  </w:r>
  <w:r>
    <w:t> formato y un </w:t>
  </w:r>
```

```
<w:hyperlink w:rel="rId4" w:history="1">
  <w:r>
    <w:rPr>
      <w:rStyle w:val="Hyperlink" />
    </w:rPr>
    <w:t>hiperenlace</w:t>
  </w:r>
</w:hyperlink>
</w:p>
```

As we can see, it would be difficult for anyone to even understand the representation of our simple text, let alone change it manually.

But the cryptic nature of EOOX is not only down to this method of encoding XML; it is also due to encodings that are specifically the result of how Microsoft's operating system, Windows, stores information in its memory. In the following example, taken from section 2.8.2.16 (page 759) of Volume 4 of the final draft of EOOX, we can see the following half XML, half binary expression:

```
<w:font w:name="Times New Roman">
  <w:sig w:usb0="2002A87" w:usb1="80000000"
w:usb2="00000008"
  w:usb3="00000000" w:csb0="000001FF"
w:csb1="00000000" />
  ..
</w:font>
```

As we can see, the hexadecimal numbers assigned to the variables "usb" and "csb" are mere bit-for-bit dumps of the Windows' memory data structure with no intervening abstraction to make them suitable for other platforms or remotely intelligible at a higher level. Didn't someone say that EOOX was a pure XML format? Hadn't we already agreed that binary formats were a thing of the Middle Ages of software development? But the worst thing is that this kind of binary, platform dependent encoding (for example, only for *big-endian* systems or only for *little-endian* systems?) is repeated for various EOOX functionalities, such as paragraph conditional formatting, table-cell conditional formatting, conditional table-row formatting, table style conditional formatting settings exception, etc. Interested readers can find more on the matter of bitmasks in EOOX in an article written by IBM professionals¹⁵.

But EOOX's dependence on a specific system and application does not end here; it is patently obvious that EOOX is not a specification of a format, but rather a journey through the history of MS-Office, an accumulation of all the patches and fixes over the last ten years. Just take a look at some of

¹⁴ Note of the Editor: This Spanish sentence means "This is a very simple and a bit formatted document, and a hyperlink".

¹⁵ An Antic Disposition, Rob Weir, "A bit about the bit with the bits". <http://www.robweir.com/blog/2006/10/bit-about-bit-with-bits.html> (last accessed 10-01-2007).

the exceptions in the specification¹⁶, such as: "*interpret this bit as X when it comes from Mac, as A when it comes from Win32 and even different when it's written before date Z*".

And there is still more. The specification requires you to take into account the operational peculiarities and exceptions (not even described in the specification and known only to Microsoft) of all the different legacy versions of MS-Office there have been, combined with the successive versions of MS-Windows in which they have run¹⁷. The chapter titles alone of some of these historical gems as they appear in EOOX's official specification should be testimony enough:

- autoSpaceLikeWord95 (Emulate Word 95 Full-Width Character Spacing) – chapter 2.15.3.6
- footnoteLayoutLikeWW8 (Emulate Word 6.x/95/97 Footnote Placement) – chapter 2.15.3.26
- mwSmallCaps (Emulate Word 5.x for the Macintosh Small Caps Formatting) – chapter 2.15.3.32
- suppressTopSpacingWP (Emulate WordPerfect 5.x Line Spacing) – chapter 2.15.3.51
- lineWrapLikeWord6 (Emulate Word 6.0 Line Wrapping for East Asian Text)
- mwSmallCaps (Emulate Word 5.x for Macintosh Small Caps Formatting)
- shapeLayoutLikeWW8 (Emulate Word 97 Text Wrapping Around Floating Objects)
- truncateFontHeightsLikeWP6 (Emulate WordPerfect 6.x Font Height Calculation)
- useWord2002TableStyleRules (Emulate Word 2002 Table Style Rules)
- useWord97LineBreakRules (Emulate Word 97 East Asian Line Breaking)
- wpJustification (Emulate WordPerfect 6.x Paragraph Justification)
- shapeLayoutLikeWW8 (Emulate Word 97 Text Wrapping Around Floating Objects)

These few examples alone should more than explain why EOOX's specification is over 6,000 pages long, not including appendixes.

Furthermore, another design problem affecting EOOX encoding is that it mixes pure content with its presentation, while OpenDocument, although it is not entirely blameless in this respect, does so to a much lesser extent and mostly keeps data separate. This separation is comparable to the way XHTML encoding uses CSS files.

Let's take another example of formatted text and the ways that each format represents it:

texto en negritas (text in bold type)

To represent it in OpenDocument, firstly the information relating to content is stored:

```
<text:span text:style-name="Enfasis_10_Negrita">
  texto en negritas
</text:span>
```

then the manner in which the content is to be presented is encoded. As we can see, the style to be used is defined by "Negrita_20_Enfasis":

```
<style:style style:name="Enfasis_10_Negrita"
  style:display-name="Enfasis con Negrita"
  style:family="text">
  <style:text-properties fo:font-weight="bold" />
</style:style>
```

Notice that OpenDocument is using "styles" to format the document. This is very useful, particularly when altering the presentation of an entire document, since it is only necessary to change the style, without having to navigate through the entire document changing the appearance of every chunk of text on a case-by-case basis.

Meanwhile, EOOX encodes this same text and format in the following manner:

```
<w:r>
  <w:rPr>
    <w:b />
  </w:rPr>
  <w:t>texto en negritas</w:t>
</w:r>
```

We can see here the odd result of mixing content with format, since <w:b /> means "bold type (*negritas*)".

To continue with our technical analysis of the two formats, if we compare the size of files and their elements¹⁸ required to store the same text of some size, we see that OpenDocument generates files that are around 90% smaller than equivalent files in Microsoft's old .doc format, and around 50% smaller than those generated by the EOOX format.

From an economic point of view, the cost of implement-

¹⁶ KDE Developer Journals, Zander, "office document formats". <<http://www.kdedevelopers.org/node/2568>> (last accessed 10-01-2007).

¹⁷ Rob Weir, "How to hire Guillaume Portes". <<http://www.robweir.com/blog/2006/01/how-to-hire-guillaume-portes.html>> (last accessed 10-01-2007).

¹⁸ O'Reilly XML.com: "Comparing XML office document formats: using XML metrics". <http://www.oreillynet.com/xml/blog/2006/08/comparing_xml_office_document_3.html> (last accessed 10-01-2007).

ing Microsoft's monstrous format specification is also far in excess of what it will cost to implement OpenDocument. An employee of the Microsoft's Mac division estimated¹⁹ that it would take 5 developer-years to program just a quarter of what was needed for just the word processor.

Meanwhile, an expert from Adobe estimated²⁰ the effort required to implement the entire specification at 150 developer-years. Finally, the VP of the OpenDocument Foundation estimated²¹ that it would cost a thousand dollars just to print out the documentation required to implement it.

Another serious problem that raises its head when implementing or adopting EOOX is that there is no guarantee that either Microsoft or any of the other members of the ECMA technical committee have declared, let alone waived, the software patents that the format infringes²². This leaves the door open to the risk of lawsuits being filed later against all the other applications that adopt the format but, of course, only once the standard is sufficiently well established, as happened in the case of Unisys and GIF²³. In fact Microsoft's licence does not cover all the patents necessary to implement the format in its entirety²⁴. Fortunately for Europeans, software patents are not legal, but the same cannot be said for the citizens and companies of other countries, starting with the USA. And obviously we need to bear in mind that the scope of international standards must be global; it makes no sense calling them "international" if they only apply to a specific region.

To conclude this comparison between the two formats, we need to take into account another, highly controversial, factor that has recently become very important. Let us suppose that the EOOX format is patent free and can be freely implemented by anyone regardless of the not insignificant difficulties that we have just described. However, up until now our analysis has ignored a key factor: DRM²⁵ (*Digital*

Rights Management or, perhaps more appropriately, *Digital Restrictions Management*). There can be no problem for Microsoft and their monopoly to open up their formats if they can then close them up again because the documents generated in their operating systems are subject to DRM unlock codes that are impossible to read by other platforms and that are actually illegal to circumvent under US²⁶ and European²⁷ laws. And Windows Vista together with MS-Office 12 fully integrates DRM and the unlock codes²⁸ that lock in the content so it can be accessed only by certain hardware and software.

Finally, we should mention the factor that most militates against EOOX, which is that after just over a year of development, compared with the five years of OpenDocument, there is currently not one application on the market that supports EOOX. Not even its sponsor and promoter, Microsoft, has a commercially available product that supports EOOX. Right at this moment it is nothing but a promise: pure *vapourware*²⁹.

4. Conclusions

The facts outlined above answer a great many questions: should we really be returning to the times when standardization bodies came up with convoluted theoretical standards that were practically impossible to meet? Or, should we be doing so merely to satisfy the commercial interests of a company that has been declared a monopoly by the authorities of the world's two largest markets? Hadn't we put that mistake behind us?

What is the point of having a specification of thousands upon thousands of pages that merely reflects the way that one particular application and one particular company does things, a specification for which there is no practical possibility of any other competitor applications supporting it? Is the idea to open up and ensure free markets in which com-

¹⁹ Effort estimation for implementing EOOX by Rick Schaut of Microsoft: "Open XML converters for Mac Office". <http://blogs.msdn.com/rick_schaut/archive/2006/12/07/open-xml-converters-for-mac-office.aspx> (last accessed 10-01-2007).

²⁰ Effort estimation for implementing EOOX by Andrew Shebanow of Adobe: "Is Office Open XML a one-way standard? Ask Microsoft". <http://blogs.adobe.com/shebanation/2006/12/open_xml_one-way.html> (last accessed 10-01-2007).

²¹ Information Week: "Microsoft's XML Standard Needs Fast Track Approval To Halt Defections", with commentary by Sam Hiser. <<http://www.informationweek.com/management/showArticle.jhtml?articleID=196602114>> (last accessed 10-01-2007).

²² Groklaw's analysis of intellectual and industrial property rights in ECMA Open Office XML. <<http://www.groklaw.net/articlebasic.php?story=20051207020812228>> (last accessed 29-12-2006).

²³ Unisys demands payment from all users of the GIF image format. Explanation from the FSF. <<http://www.gnu.org/philosophy/gif.html>> (last accessed 29-12-2006).

²⁴ Sam Hiser, "Analyzing the Microsoft Office Open XML license". <http://fussnotes.typepad.com/plexnex/2007/01/analyzing_the_m.html> (last accessed 15-01-2007).

²⁵ Wikipedia, DRM. <http://en.wikipedia.org/wiki/Digital_Rights_Management> (last accessed 28-02-2007).

²⁶ US, Digital Millennium Copyright Act (DMCA) analysed by the EFF. <<http://www.eff.org/IP/DMCA/>> (last accessed 10-01-2007).

²⁷ EU Directive 2001/29/EC "on the harmonisation of certain aspects of copyright and related rights in the information society" according to Wikipedia. <http://en.wikipedia.org/wiki/Directive_on_the_harmonisation_of_certain_aspects_of_copyright_and_related_rights_in_the_information_society> (last accessed 10-01-2007).

²⁸ Jeff Farris, "Remote Attestation". <<http://www.math.uiuc.edu/%7Eduursma/Math595CR/FarJ.pdf>> (last accessed 10-01-2007).

²⁹ Daniel Eran, "Microsoft's Yellow Road to Cairo". <<http://roughlydrafted.com/RD/Q4.06/4E2A8848-5738-45B1-A659-AD7473899D7D.html>> (last accessed 15-01-2007).

petition lowers prices and drives innovation or, conversely, is it a matter of locking everyone into a single application driven by the commercial interests of a monopolistic manufacturer thanks to its patents, DRM, whopping specifications, etc.?

As we have seen in the last point of this article, EOOX doesn't even have any technical advantages over ODF. In fact, as is mentioned on the *OpenDocument Fellowship* website, we have yet to find a single thing that EOOX can do that OpenDocument cannot. Not one.

Finally, in this day and age it makes no sense for the public authorities to consider having two standards when practically nobody, with the exception of one company in particular, will benefit in the slightest from such a measure. It would be like reintroducing Turkish miles just so the metric system had some competition. Is the metric system in need of competition? I think not.

Official open standards that do the job required of them, as OpenDocument does, should not have, nor do they need, unnecessary alternatives; on the contrary, they should be demanded and promoted by the public authorities. Where there should be competition and freedom of choice is between manufacturers and their applications, not between standards of more than sufficient quality. What is the point of causing errors, chaos, and confusion where there is a magnificently productive harmony?

All these conclusions are drawn in the context of a situation in which OpenDocument has the backing of practically all the office automation market. This is the best reflection of what future electronic document formats for Humankind: open, universal, and free standards, like the alphabets, languages, and paper making techniques of the past which have allowed us to benefit from the wisdom of our forebears.

Meanwhile, ECMA/MS-Open Office XML's only contribution is to prolong unnecessarily a history of users being locked into the formats and applications of a series of single companies, regardless of the greater or lesser market share they may have had or whether or not they were monopolies at the time. Thus the possible acceptance of EEOX as an international standard and its success on the market can only mean a return to the Middle Ages of electronic documentation. To continue living in an age in which Humankind suffered the hijacking and censorship of knowledge at the hands of a few feudal lords with special rights that are denied to the rest of us mortals. In short, possibly one of the worst obstacles to the development of the Information Society that we have ever known.

In short, **software should be designed to meet the standard**, not the other way round. **Humankind already has a document standard**. Its name is ISO 26300, **OpenDocument**.

Acknowledgements

My thanks go to Alex Hudson, J. David Eisenberg, Bruce D'Arcus, Daniel Carrera and Marco Fioretti (OpenDocument Fellowship), to Rob Weir and Bob Sutor (IBM), to Simon Phipps

and Tim Bray (Sun Microsystems), to Marino Marcich (ODF Alliance), Sam Hiser (OpenDocument Foundation), to Rafael Rodríguez (*Universidad de Cádiz*) and any others that I may have inadvertently left out, without whose tireless efforts in the study and promotion of open document formats this article would not have been possible.

Translation by Steve Turpin