Comparative assessment of Open Documents Formats

Market Overview

Document administration

Current version

Reference	Version	State	Exploitation	Sender
20033012.MKT.ODF.1.6.doc	1.6	Draft		A.Rizk

Written by	Checked by	Validated by
VALORIS		A.Rizk

List of addressees	Copy for	
	Application	Information
• BERNHARD SCHNITTGER (EC)	X	
• EMILIO CASTRILLEJO (EC)		X
• VEMUND RIISER (EC)		X
• ANTOINE RIZK (Valoris)	X	
• MICHEL TUENI (Valoris)	X	

Chronology of modifications

Follow-up of modifications			
Version/Revision	Date	Modifications	Origin
1.0		Creation of the document	VALORIS
1.6		Following EC remarks	VALORIS

Disclaimer

"The views expressed in this document are purely those of the writer and may not, in any circumstances, be interpreted as stating an official position of the European Commission.

The European Commission does not guarantee the accuracy of the information included in this study, nor it accepts any responsibility for any use thereof.

Reference herein to any specific products, specifications, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favouring by the European Commission.

All care has been taken by the author to ensure that he has obtained, where necessary, permission to use any parts of manuscripts including illustrations, maps, and graphs, on which intellectual property rights already exist from the titular holder(s) of such rights or from his or their legal representative."

TABLE OF CONTENTS

Executive Summary

The purpose of the work described in this document is to evaluate market trends and assess existing or emerging open document formats (ODF) as a step towards recommendations on their use for office documents exchanges between EU member states' administrations.

The current landscape of ODF is crowded with legacy and normative ODF. After an exhaustive panorama of open document formats that apply to office documents, a selection of criteria were applied in order to establish which ones are worthy of detailed analysis. These criteria are technical and non-technical. They include characteristics such as openness, rewrite-ability, format fidelity, interoperability, adoption etc. Although no single format meets all of the criteria, two formats were further analysed namely, Microsoft XML reference schemas and OpenOffice.org.

After increasing pressure from users and EU Administrations, Microsoft has announced on 17th Nov. 2003, the publication of the XML Reference Schemas for its office suite. Except for some technical details, these XML formats seem to be completely documented, and licence to use them is granted freely. However, the licence includes some constraints, which need to be examined carefully as they might be too binding and incompatible with GPL software integration.

OpenOffice.org (OOo) is both an office applications suite and an XML based format. OpenOffice.org is a community-based project and is based on the open-sourced code from an older version of StarOffice bought by Sun Microsystems from German software company StarDivision, founded in 1991. In 2000, Sun released the source code of StarOffice software publicly through OpenOffice.org, thus initiating the world's largest open source project.

OpenOffice is currently being standardised by Oasis. The chairman of the Oasis technical committee in charge of OpenOffice predicts that the standard will be voted in the first half of 2004. OOo boasts a confirmed and growing user base estimated between 10 and 40 million users, mainly in governments and administrations. In its attempt to standardize the OpenOffice.org format, Sun Microsystems is not backed up by other market players except for the Open Source community.

This document argues that Microsoft has a clear technological lead, especially regarding the possibility of creating and processing documents according to user-defined schemas in addition to the XML reference schemas, which allows seamless integration of documents in other parts of the information system. However, given the other advantages of OOo (open, multiplatform, multilingual, free or low-price for StarOffice), a great number of users will be satisfied with the functionalities of OOo, which according to analysts will leave it with a potential of 10% market share.

Our view is that the corresponding formats, namely MS XML Reference schemas and OOo will naturally follow the adoption of the major tools behind them. Microsoft's XML lead and market dominance will remain for the few years to come. On the other hand, OpenOffice user base size is now such that it is irreversible, and it constitutes a viable alternative to Microsoft. In terms of wide adoption of the format, we believe that none of the two will be winner or a knockout looser, with MS dominating the user base at 85%. The two formats will coexist, but OOo will become more and more the open format reference for interoperability across platforms.

1 Introduction

1.1 Context and main objectives

By decision of the European Parliament and Council, The IDA Programme was set up in order to encourage information content interoperability through the promotion of trans-European telematic networks between Administrations, Institutions and Agencies.

As e-government is developing, the need for automatic application-to-application interchange, based on shared protocols and business document definition, is growing. Nevertheless, the non-automatic exchange of documents between individuals is still one of the most common types of interaction. If the exchange of documents through email (or HTTP) has become common practice, the interoperability of document formats can still be an issue.

The purpose of the work described in this document is to evaluate market trends and assess existing or emerging open document formats as a step towards recommendations on their use for office documents exchanges between EU member states' administrations.

1.2 Our methodology

In order to obtain a clear picture of the ODF market, both in terms of players' IT strategy, technical evolution of formats, and vendor plus user momentum, we have proceeded according to the following steps:

- Establish an exhaustive list of currently available document formats, and describe them succinctly in order to provide the initial starting material. Much of this work was inspired and synthesised from a previous study on the subject provided by the BSI group (Bundesamt für Sicherheit in der Informationstechnik).
- Establish a set of criteria, both technical and non-technical, that are fundamental to a proper assessment of ODF.
- Given the criteria, select potential candidates for a more detailed study:

  - XML Reference Schemas from Microsoft, OpenOffice.org from SUN initially, then OpenOffice.org and Oasis, and to a lesser extent XML+PDF from Adobe.

- For the above formats:
  - Read and analyse technical evaluation and comparative studies
  - Interview main players
  - Collect information sources (documents, websites etc.)
  - Obtain market figures (Forrester, Giga, Ovum etc.)

The main players interviewed and contacted are as follows:

  - Adobe
  - Arbortext
  - IBM
  - Koffice
  - Microsoft
  - OASIS
  - SAGE
  - Software AG
  - SUN

Attendance at two major events on the topic has also provided valuable input:

- SUN's seminar on StarOffice at Integration XML, Paris Nov. 2003
- Aristote seminar on XML office document formats, Paris Dec. 2003.

A list of major websites visited and bibliographic references is given in the annex.

1.3 Structure of the document

The present document is dedicated to the description of our main findings, and follows the order of our stepwise methodology described above, namely:

Chapter 2 enumerates and briefly describes the currently available office document formats. It enumerates and describes the selection criteria, and concludes by choosing two document formats that qualify for further evaluation, namely OpenOffice.org and MS XML reference schemas.

Chapter 3 and 4 are dedicated to a detailed analysis of each of the two selected formats respectively, and so from a historical, technical and market perspective.

Chapter 5 briefly describes the technical functionality of digital signature in Open Document Formats and processing tools. This functionality was deemed important to zoom on, given that it has been overlooked from common technical comparative studies on the topic.

The principal conclusions are finally summarised, with a note on how we interpret the current market situation and its future trend.

2 Office Document Formats

2.1 Formats and document representation

Word processing programs are used to create and format documents.

In reality word processing tools work on an abstract internal representation of the document, and use a document format to save and exchange it with other tools. Although some document formats are meant to be independent of a given tool, in practice they almost all reflect and follow the functionalities provided by a word processing or viewing tool. Their evolution also follows closely the arrival of new versions of the corresponding "parent" word processor.

Some document formats indeed work with many tools, especially those that are meant for viewing such as HTML and PDF.

2.1.1 Internal representation

Although the representation of the text is the primary focus, most word processors do not work directly with the print image for their internal representation. Instead, they manage character sequences entered by the user (characters, words, paragraphs, chapters, pages, etc.), assign mark-up to these objects (indented, bold, italics, Times, 16pt, etc.) and produce an on-screen image from this internal representation in real-time. The WYSIWYG (What You See is What You Get) is achieved because the algorithms that produce the on-screen image produce very similar results as the algorithms that later put the ink on the printout.

Modifying a text document at the level of the print image is much more difficult than modifying text and attributes. This is why word processing programs do not work with the print image but rather with the text that forms the basis of the print image and its associated attributes. Changes to documents are always made at this abstract level.

Just like any other program, a word processing program works with many variables and data structures in defined and dynamically allocated storage areas. The status of the programs and the text along with its associated properties are saved in the computer's main memory. All of these are part of the internal representation and by consequence, must be stored in order to reconstruct the internal representation from file.

2.1.2 Document format

When a text document is saved by the program into a file, the complex internal representation of the document has to be written down from the main memory of the computer to the hard drive in such a way that the document can be reconstructed from this file at a later time.

The easiest way to store this internal representation would be a complete storage dump (ie. direct load of main memory state to file system). When this dump is reloaded into the main memory from the file, the internal representation is again available. In this case, the stored data is only significant to the program that wrote it, and results in a proprietary document format.

Some proprietary document formats have implemented a minimum level of functionalities assuring the document can be opened on different computers. They however remained incompatible between versions (e.g. different Word versions) and platforms (e.g. Mac and PC).

To solve the issue of incompatibility, word processors provide import and export filters, which allow to import and export documents to other formats. Newer versions of the software tend to be backwards compatible. However, as technology evolves word processors offer more functionality resulting in formatting information loss, and hence format degradation, if documents are converted to an older format. This situation increases the complexity for the end-users and is certainly not beneficial for easy exchange of documents in an increasingly networked and collaborative business environment.

It is therefore now increasingly important to standardise the document format by publishing its specifications and making them available. When the file is saved, the internal representation of the text document is converted to the standard format. Inversely, when the file is read by the same or by another tool, the format is abstracted and converted to the internal representation.

2.1.3 Document format components [i]

Electronic documents contain many types of components that can be very different from one another. In addition to text, the user may want to integrate various types of graphics, display diagrams or add spreadsheets. Some of the components that a digital document is broken down into are already well standardised. For example, pixel graphics and pictures can exist as JPEGs, GIFs or PNGs.

It is also necessary to manage data related to the main document that are not actually part of the content. These data could include, for example, information about the configuration of the program or format templates.

There is also a growing trend to store document metadata. Metadata is information about the document itself. Examples are the author, title, subject, keywords, revision number, editing time, number of words, number of paragraphs, etc.

So in a given document, we can identify 5 types of information, which should therefore be catered for in the document format:

- The textual content itself
- Markup information
- Multi-media components
- Document meta data
- Data related to the configuration of the originating application

2.1.4 The XML standard

2.1.4.1 Background

XML (eXtensible Markup Language) was developed by an XML Working Group (originally known as the SGML Editorial Review Board) formed under the auspices of the World Wide Web Consortium (W3C) in 1996. It was chaired by Jon Bosak of Sun Microsystems with the active participation of an XML Special Interest Group also organized by the W3C, and that includes players such as Microsoft, Arbortext etc

XML makes use of only a few associated standards, namely Unicode and ISO/IEC 10646 for characters, Internet RFC 1766 for language identification tags, ISO 639 for language name codes, and ISO 3166 for country name codes.

XML is used for the description of marked-up electronic text. More exactly, XML is a *metalanguage*, that is, a means of formally describing a *mark-up* language. A mark-up language specifies what mark-up is allowed, what mark-up is required, how mark-up is to be distinguished from text, and what the mark-up means. XML provides the means for doing the first three; supplementary documentation is required for the last.

An XML based document format is described by an XML mark-up language.

2.1.4.2 Main characteristics

There are three main characteristics to XML:

1. emphasis on descriptive rather than procedural mark-up

In a descriptive mark-up system, the mark-up codes or tags used do little more than categorize parts of a document. They simply define what the element content is rather than how it is processed. In XML, the instructions needed to process a document for some particular purpose (for example, to format it) are sharply distinguished from the descriptive mark-up, which occurs within the document. They are collected outside the document in separate procedures or programs, and are usually expressed in a distinct document called a *stylesheet*, though it may do much more than simply define the rendition or visual appearance of a document.

There are at least two main advantages to this separation of description from processing:

- The same document can be rendered or processed differently on multiple channels, and for multiple user profiles.
- The document can be archived without it being encumbered by machine dependent processing instructions.

2. The *document type* concept

The type of a document is formally defined as a DTD (Document Type Definition) or as an XML-Schema. The document type specifies what elements are allowed in a document instance, and how the elements can be ordered, with constraints on how many times they can occur. In addition, with XML schema one can specify the type of the element content.

If documents are of known types, a special purpose program (called a *parser*), once provided with an unambiguous definition of a document's type, can check that any document claiming to be of a that type does in fact conform to the specification. A parser can check that all and only elements specified for a particular document type are present, that they are combined in appropriate ways, correctly ordered and so forth. More significantly, different documents of the same type can be processed in a uniform way. Programs can be written which take advantage of the knowledge encapsulated in the document structure information, and which can thus behave in a more `intelligent' fashion.

3. Independence of any one hardware or software system

A basic design goal of XML is to ensure that documents encoded according to its provisions can move from one hardware and software environment to another without loss of information. All XML documents, whatever language or writing system they employ, use the same underlying character encoding This encoding is defined by an international standard, which is implemented by a universal character set maintained by an industry group called the Unicode Consortium, and known as Unicode.

2.1.4.3 Facts about XML

The mark-up language with which XML is most frequently compared is HTML. With respect to HTML, XML has some other important characteristics:

ü XML is extensible: it does not contain a fixed set of tags

ü With XML one can define an infinite number of DTD or document types. HTML is *one* such DTD.

ü XML focuses on what data is, not on how it is presented as a web page.

What makes XML unique is that it can represent unstructured, semi-structured, and structured data with equal ease. XML does not specify any particular manner for how the data should be processed, handled or presented.

The capability of XML to separate its process and data content provides it with the capability to future-proof encoded content.

While XML provides the mechanism to separate data from processing content, in experience, it is up to the human designer to make sure that this separation actually occurs[ii]. It is quite trivial to add elements to an XML document that place processing requirements and restrictions on the document, thus preventing cross-platform processing capability. In fact, the potential for creation "proprietary" XML document formats has many in the Open source community concerned.

While properly developed XML should in theory be platform-neutral, experience has shown that vendors who wish to maintain and protect their platform's market will go to extents to encode elements that are capable of being processed only by their own application suites. The only counter-balance to this natural force is the development of open, cross-industry, widely adopted standards that serve to block the inclusion of application or platform specific encoding.

Moreover, documents that obey to different XML based formats or DTDs are not necessarily compatible. Conversion between the two formats could even prove extremely difficult, or even impossible.

XML has now become the *lingua franca* for data exchange between information systems. Companies are realising significant improvements and optimisation of their processes by adopting XML as a core foundation for the content lifecycle. However, no ROI at all can be gained from an XML enabled content lifecycle if the content itself is not represented in XML

Unfortunately, the major hurdle for using XML is that most authors do not get any benefit from converting to structured authoring. Whilst improving the content lifecycle, and providing high ROI for an organisation, encoding content in XML is not directly rewarding for the authors who create the content. Unless imposed by the organisation, authors usually reject adding XML structure to the content.

2.1 Past European experience: ODA

Having a general standard open document format is not a new aspiration [iii]. This dates back to early 80's as electronic office documents became commonplace. Europe had at the time its own initiative for defining and standardising a common format called *Open Document Architecture* (ODA). ODA received a lot of European institutions backing, as well as funding from European Commission. ODA was originally developed by the European Computer Manufacturers Association (ECMA) who published the standard as ECMA 101 in 1985. It was republished in 1989 as ISO 8613 Office (now Open) Document Architecture (ODA) and Interchange Format; it was also published by the ITU in 1989 as its T.410 series of recommendations. The ISO and the ITU issued identical revised versions of the Standard in 1994 and 1993 respectively.

In the words of the Standard, the purpose of ODA is "To facilitate the interchange of documents in a manner such that:

- Different types of content, including text, image, graphic and sound can coexist within a document
- The intentions of a document originator with respect to editing, formatting and presentation can be communicated most effectively."

ODA sought to provide a global standard for describing and handling documents that is independent of any proprietary format or equipment. The principle is that if every supplier uses this architecture as the basis for their document handling systems, the networking and archiving problems *http://www.nhsia.nhs.uk/napps/step/pages/ithandbook/h235-1.htm - h235DEX*will be eliminated.

The widespread use of ODA was expected to provide an efficient method of exchanging documents between different applications

ODA unfortunately failed, and gave birth to little spin-off results. The main reasons of this failure can be summarised as follows:

- The standard was too ambitious for the computer capabilities at the time;
- It was too complex to adopt, and remained theoretical with no tool support;

- The arrival of more pragmatic standards such as SGML shifted support away from ODA;
- The generalization and evolution of RTF from Microsoft;
- The standard grew out of control;

Furthermore, ODA had some technical drawbacks, namely:

- The editing structure could not be preserved between many word processing applications;
- ODA did not provide the same set of primitive descriptions of data as many other systems
- ODA was highly redundant in that several features can be used for representing similar features

The natural consequence of the first two deficiencies is that many translators from native formats to ODA had to save their extra-ODA information using particular structures and encodings of ODA objects.

The third deficiency is asymmetric. When generating ODA format, a translator could minimize the third deficiency by picking one of the possible representations in ODA. However, if a translator from ODA format to a native format has no a priori knowledge about the generator of the ODA file, the translator could not assume that any particular conventions were followed. Thus, the translator would have to deduce the particular way that ODA was used to encode features that have multiple representations.

2.2 List of Criteria

The “ideal” office document format should be:

- Open
- Non-binary
- Modifiable
- Preserve format fidelity
- Support current word processor features
- Support emerging requirements
- Widely adopted

These criteria cannot be seen independently. Cross-platform interoperability and extensibility require the document format to be non-binary. Saying the document format should support all currently available word processor features requires that the document format be modifiable.

This section describes each one of the above criteria.

2.2.1 Open

The minimum requirements for an open standard are that the document format is completely described in publicly accessible documents, that this description may be distributed freely and that the document format may be implemented in programs without restrictions, royalty-free, and with no legal bindings.

2.2.2 Non binary

The document textual component with its corresponding mark-up can either be saved as a binary data stream (Microsoft Word, PDF) or in plain text format (all XML formats and RTF).

The standard should be non-binary as there are important drawbacks associated with binary formats:

- **Platform dependency**: Whether the documents can be accessed and used is dependent on the software and platform. For instance, Microsoft Word files cannot be exchanged between Intel and Mac independently of architecture. The availability of the hardware and software is in no way guaranteed in the long run. As such binary file formats may not be compatible with the tools of the future.
- **Increasing awareness for long-term archiving:** Just as for paper documents, when storing electronic text documents, there are certain mandatory storage periods than must be conserved. This aspect is gaining importance with the steady increase in digital communication and the trend towards a “paperless office”. Currently, the market solves this problem by supporting many different document formats. By converting old formats to newer ones, it is theoretically possible to access data at all times. Considering the rapid change in document formats, there is no satisfactory solution available based on proprietary document formats for documents that are required to be archived (for up to 60 years in government agencies).

2.2.3 Modifiable

We must separate modifiable document formats from document formats intended only to distribute information. Modifiable formats allow further modifications by one or more recipients, often part of a collaborative project. In contrast with these formats are those formats intended for sharing information, not modifying information.

Documents with this last type of document format are used only to transmit the contents of a document, keeping the format intact. The document is merely an electronic representation of the printed out document. Portable Document Format (PDF) for instance, is designed as a format for sharing a fixed printed image. It offers format fidelity in both the on-screen display as when printed, but does not allow to modify information.

2.2.4 *Preserve format fidelity*

Documents can lose meaning and value if the layout or visual emphasis is altered. In legal or regulated environments, retaining fidelity may be a requirement for official communications or legally binding transactions.

Lack of presentation fidelity could also have dangerous effects. On June 27, 1988, a train crashed into the Gare de Lyon in Paris at high speed and without braking. More than fifty people were killed in the accident and a considerable number wounded. The accident was found to be caused by the coincidence of two badly formatted indented instructions in the maintenance manual [iv].

We define **presentation fidelity** of a document format its ability to preserve the original layout of the document, regardless on which platform or computer the document is opened.

Other aspects of fidelity are **content** and **structural fidelity**. As content fidelity is an absolute condition sine qua non, this is not a discriminating factor and thus, not a criterion.

2.2.5 *Cross-platform interoperability*

Cross-platform interoperability implies that the format can be exploited, with full preservation of its semantics on various hardware (PC, SUN, Mac..) and software platforms (Windows versions, Linux and Unix versions under different desktop environments etc..).

This criterion is related to that of fidelity inasmuch as the semantics of the format cover presentation and structure fidelity. Fully preserving the semantics in this case implies respecting fidelity.

By definition, only non-binary document formats, although not necessarily all of them are able to comply with this criterion.

2.2.6 Support for current word processor features

The document format should be able to represent common features found in currently available word processor applications:

- WYSIWYG editing
- Tables, multiple columns …
- Revision marks
- Scripting
- Support for UNICODE character sets
- Bi-directional (Hebrew and Arabic language) and vertical writing (Asian languages)
- Etc.

2.2.7 Support for emerging word processor features

This criterion extends the above one to those more advanced features, which we believe will be part of the future word processor tools:

- Digital signatures
- Access rights
- Copyright and Digital Rights Management
- Version control and Collaboration
- Support for user-defined XML Schemas. As discussed earlier, we foresee electronic document to become more and more decoupled from application. More and more applications will intervene on subparts of the document. The recently added support for user-defined XML schemas in Word2003 confirms this trend. A Word document can now contain Word mark-up, in combination with extra attributes to be used by other business applications. Situations are imaginable where many applications intervene in a similar way on the same document.

2.2.8 Widely adopted

The ODA experience described earlier proves that there is little interest in having a theoretical format that is complete on paper, if no tools implement it. The W3C in this regard, does not produce a recommendation unless at least one tool exists that implements it already.

Wide adoption does not necessarily imply market dominance or being **the** universally accepted document format.

By wide adoption we mean that there is sufficient user and tool vendor momentum to sustain the format's existence and exploitability. This is a market dynamics issue that can be influenced by the technical aspects of the format, but is mainly dependent on market factors such as vendor strategies, and user policies.

2.3 Current landscape

2.3.1 Overview

The following table gives an overview of the most common document formats together with the main “parent” tool that uses them as a native format.

Looking at the current landscape of office document formats, we can make following observations:

- There are many formats available. This fact increases the need for an ODF that could act as a pivot for cross-tool interoperability. Otherwise, each tool would have to create import-export filters to all available formats, i.e around 400 different combinations for 20 tools/formats. In addition to problems with cost and maintenance of such tools, the risk of incompatibility and error is at its maximum.
- Many of the formats come from Microsoft. Despite this fact, compatibility between the various MS formats is far from being guaranteed.
- There is a close, often one-to-one dependence between each format and its parent tool. A format cannot therefore be analysed in full abstraction from its parent tool and other available tools that use it. Whenever the format’s semantics leaves room for interpretation, the behaviour of the parent tool with the format is usually considered as the reference.
- XML in itself is NOT a format in its own right. XML only enables the definition of formats. It is therefore not part of the list. Many formats are based on XML and yet they are not interoperable. This is due to the fact that they obey to different and complex DTD’s or Schemas.
- The list does not include vertical formats such as ATA100 and AECMA for technical documentation, TEI for literary documents, etc. We have decided to limit our attention to more horizontal office document formats.

This section describes very briefly each of the above formats, in the light of the ODF criteria.

Document Format	***Abbreviation***	***Tool***
WordProcessingML SpreadsheetML FormTemplateML	WordML	Micosoft Office 2003
Word 2002	doc2002	Microsoft Word 2002 XP
Word 2001	doc2001	Microsoft Word 2002
Word 2000	doc2000	Microsoft Word 2002
Word 97	doc97	Microsoft Word 97
Word 95	doc95	Microsoft Word 95
Word 6 for Windows	doc6	Microsoft Word 6 for Windows
Word 2 for Windows	doc2	Microsoft Word 2 for Windows
Rich Text Format	rtf	Many word processing programs
Word 5.5 for Dos	doc55dos	Microsoft Word 5.5 for MS-Dos
WordPerfect9	wp9	Corel WordPerfect 9
WordPerfect10	wp10	Corel WordPerfect 10
(La)TeX	latex	Text editor
StarWriter 5.2	sw52	StarOffice 5.2
OpenOffice.org 1.0	OOo1	OpenOffice.org 1.0
		StarOffice 6.0
		OpenOffice.org 1.1
		StarOffice 7.0
Kword 1.2	kw1	Kword 1.2
AbiWord 2.0	aw2	AbiWord 2.0.1
FrameMaker 7.1	fm7	Adobe FrameMaker 7.1
Portable Document Format	pdf	Acrobat Distiller 5.0, etc.
XML Data Package	xdp	Adobe
Postscript	ps	Printer driver and Ghostscript, etc.
Device-independent	dvi	(La)TeX
XHTML	html	multiple
DocBook	docbook	multiple

2.3.2 Microsoft DOC format[v]

The native document format of Microsoft Word follows the internal program representation very closely; using Microsoft's Object Embedding and Linking technology. A Word DOC file is OLE version 2 (OLE2) compatible, which is saved as a set of data streams in hierarchical archives. The data format of these archives is dependent on the architecture. For this reason, Microsoft DOC files cannot be exchanged as such, between Intel i386 and Mac architectures.

There are multiple binary data streams contained in the OLE2 archive:

- **Summary information stream** contains summary and meta data
- **Main stream** contains the text and the formatting information
- **Table stream** contains coded references in table format between all data structures of data stream and summary information stream.
- **Data stream** contains graphics and all embedded objects in their native format. These data are not touched by Word but manipulated by the respective OLE application embedded in Word.

Microsoft DOC format is a proprietary Microsoft standard. Documentation on the structure of the Microsoft Word file format is only available up to version 97. The newer versions of Word do not offer a separate filter for saving in Word 97, which leads to the assumption there are no essential evolutions in the format. Nevertheless, there are differences in file size observed. Word 2003 changes the size of the test files originally created in Word 2000 when they are opened and saved in the doc format.

2.3.3 RTF[vi]

The RTF format was created by Microsoft in the mid '80s to make a uniform text exchange format with graphic integration available. The RTF format has been expanded along with each new version of Word. The original version 1.0 has become today's version 1.7. RTF was however not as successful as a standard exchange format for documents requiring further modifications, as PDF was for sharing information. Today, documents are usually exchanged in DOC format.

RTF is the intellectual property of Microsoft. It adheres and evolved with MS Office versions. Microsoft provides technical documentation of RTF, but there exists no normative document that specifies it.

RTF is a text format. An RTF file is made up of control words, control characters and unformatted plain text. Because RTF has always been designed as a format to deal with different operating systems and output devices, all control commands are saved in 7-bit ASCII. The remaining character set is not defined so that ANSI, MAC and PC character sets can be used. Version 1.6 and higher have been expanded to allow for display of Unicode, which is an important criteria for multilingual interoperability.

RTF cannot save macros, which makes it invulnerable for macro viruses. RTF does not support password protection or encryption. Embedded graphics are not compressed, which is perceived by the user community as an important drawback. RTF suffers the same problems related to backward compatibility as Word and problems with tabs and tabular content (rows and columns).

XML reference schemas of Microsoft have recently replaced RTF for the future versions of MS Office.

2.3.4 TeX / LaTeX[vii]

TEX (usually written TeX in plain text) is a **typesetting system** written by Donald Knuth, which is popular in academia, especially in the mathematics, physics and computer science communities.

- LATEX (usually written LaTeX in plain text) is a document preparation system for the TeX typesetting program. It offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting .

LATEX was originally written in 1984 by Leslie Lamport and has become the dominant method for using TeX; few people write in plain TeX any more. The current version is LaTeX2•. . LaTeX is the de facto standard for the communication and publication of scientific documents.

TEX / LATEX has never expanded its adoption beyond the closed scientific community, and publishing professionals. This is basically due to 3 main reasons :

- TeX was designed as a format, for which tools were to be developed later. These tools were too long to appear, and their ergonomy was difficult to accept by an ordinary office worker.
- The gradual arrival of SGML tools made publishers and certain professional technical writers shift to them. Certain SGML tools such as “The Publisher” from Arbortext kept TeX as a hidden end format. They later removed TeX altogether.
- TeX kept its dominance for mathematical formulae typesetting. The arrival of MathML however, and its implementation in recent tools such as OpenOffice will make people shift away from TeX even more.

2.3.5 PostScript[viii]

PostScript is a programming language optimised for printing graphics and text (whether on paper, film, or CRT is immaterial). it is a **page description language**. It was introduced by Adobe in 1985 and first appeared in the Apple LaserWriter.

The main purpose of PostScript was to provide a convenient language in which to describe images in a device independent manner. This device independence means that the image is described without reference to any specific device features (e.g. printer resolution) so that the same description could be used on any PostScript printer without modification.

PostScript is an end-form non re-writeable format.

2.3.6 XHTML[ix]

XHTML stands for Extensible HyperText Markup Language, and is a reformulation of HTML 4 as an XML 1.0 application. HTML 4 is an SGML (Standard Generalized Markup Language) application conforming to International Standard ISO 8879, and is widely regarded as the standard publishing language of the World Wide Web. SGML is a language for describing markup languages, particularly those used in electronic document exchange, document management, and document publishing. HTML is an example of a language defined in SGML.

HTML4 has been designed with the help of experts in the field of internationalization, so that documents may be written in every language and be transported easily around the world. This has been accomplished by incorporating [RFC2070] [x], which deals with the internationalization of HTML.

One important step has been the adoption of the ISO/IEC:10646 standard (see [ISO10646] [xi]) as the document character set for HTML. This is the world's most inclusive standard dealing with issues of the representation of international characters, text direction, punctuation, and other world language issues.

HTML gives authors the means to:

- Publish online documents with headings, text, tables, lists, photos, etc.
- Retrieve online information via hypertext links, at the click of a button.
- Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.
- Include spread-sheets, video clips, sound clips, and other applications directly in their documents

HTML has been developed with the vision that all manner of devices should be able to use information on the Web: PCs with graphics displays of varying resolution and color depths, cellular telephones, hand held devices, devices for speech for output and input, computers with high or low bandwidth, and so on.

For this reason, a great deal of formatting is done by the web browsers designed for displaying and hyperlinking web pages. HTML has never been designed to have full control over how the document is actually displayed and paginated. By absense of built-in mechanisms, with the advent of the world wide web, web designers have extensively exploited tables and images as a means for laying out pages.

XHMTL has non-negligable advantages like true cross-platform support, it is an open standard, it is text based and it allows further modification.

Multimedia components are generally not embedded but referenced, although this is technically possible. The major drawback of XHTML is that it scores very poor as for format fidelity. The way a document looks is influenced by web browser and browser version, locally installed fonts[1], screen size settings and user preferences (user can modify text size). By "exploiting" tables and images, the format fidelity can be improved. However, this complexifies the documents and may prohibit further modification of the document.

Given the above remarks, XHTML cannot be considered as a general purpose office document format, rewiteable with sufficient fidelity.

2.3.7 MS XML Reference Schemas[xii]

Microsoft Corp. has announced on November 17th the availability of Open and -Free License for the Office 2003 XML reference schemas xiii. Microsoft has first delivered the WordprocessingML schemas; the other schemas namely, SpreadSheetML and FormTemplateML, have followed on December 5th.

These schemas describe how information is stored when documents are saved as XML. By making these schemas available royalty free, Microsoft wishes to attain a "new level of transparency, interoperability, document portability and ease of communication".

The documentation seems complete which was not the case for the previous WordProcessingML versions.

In addition to WordProcessingML, Microsoft Office Word 2003 also includes support for custom XML schema definitions[2] (XSDs), making it possible to attach one or more custom schemas to a given Word document. It allows the users to annotate the document with the elements found in the attached schemas. This makes it possible to inject business-related markup into the documents so documents can be processed around business markup instead of the more generic WordProcessingML markup.

2.3.8 OpenOffice.org 1.1 / StarOffice 7.0[xiv]

[1] There are only 4 universal fonts: arial, verdana, times new roman and courier new

[2] Schemas must adhere to the World Wide Web Consortium (W3C) XML Schema recommendation.

The OpenOffice.org format is an XML based format, which is fully documented and freely available from the OpenOffice.org open source community. Its use and extensibility is provided freely with no legal constraints.

OpenOffice was initiated by Sun Microsystems who put its StarOffice suite in the open source domain, and started the standardisation activities around its format via OASIS, the “Organization for the Advancement of Structured Information Standards” [xv].

The purpose of the OASIS OpenOffice Technical Committee is to create an open, XML-based file format specification for office applications. The resulting file format must meet the following requirements:

- It must be suitable for office documents containing text, spreadsheets, charts, and graphical documents,
- It must be compatible with the W3C Extensible Markup Language (XML) v1.0 and W3C Namespaces in XML v1.0 specifications,
- It must retain high-level information suitable for editing the document,
- It must be friendly to transformations using XSLT or similar XML-based languages or tools,
- It should keep the document's content and layout information separate such that they can be processed independently of each other, and
- It should 'borrow' from similar, existing standards wherever possible and permitted.

A lot of emphasis was placed incorporating existing W3C standards and recommendations. This is why the format includes elements and attributes of HTML, XSL-FO, Xlink, Dublin Core, MathML and SVG.

OASIS believes the format will be voted as a standard during this first semester 2004.

2.3.9 *KWord*[xvi]

KWord is part of the Koffice Suite. TheXML DTDs are freely available and they are open which means they can also be modified by third parties. The current version is Koffice 1.2.1. Up to KOffice 1.3, the XML tags used by KOffice's applications will be private to KOffice.

The KWord 1.2 package format is a TGZ (GnuZipped Tar archive) which contains at least three files:

- **maindoc.xml:** This file contains the bulk of the KWord text, tables and formula information. It is marked with XML™ tags according to the official KWord 1.2 DTD is located at: http://www.koffice.org/DTD/kword-1.2.dtd.
- **documentinfo.xml:** This file contains the document information. This is information entered into the dialog boxes when selecting File->Document Information from the menubar. This information is useful for tracking authors, contact information etc. The DTD for KOffice 1.2 is located on their web site[xvii].
- **Mimetype:** This file contains the mimetype for KWord files. This information is used by KDE to determine that this is a KWord file. This file always contains: application/x-kword

In addition, there may be other files included in the KWord document file. Pictures, embedded documents and other binary information are stored within the KWord document as separate files.

Unlike other word processing programs, KWord uses a frame-oriented layout rather than a page-oriented one. All content is created in frames whose size and position can be adjusted. The content flows between the frames unless there is a fixed frame break. The program has however, a desktop publishing mode and a word processing mode. Very often only a very simple layout is needed. The document format is the same in both modes.

As from KOffice 1.4, the plan is to switch to OpenOffice.org file formats as the new native format for KOffice.

2.3.10 *AbiWord*[xviii]

AbiWord is a free word processing program similar to Microsoft Word. AbiWord is part of a larger project known as AbiSource. The goal of the project was the development of a cross-platform, Open

Source office suite beginning with AbiWord, the project's word processor. Version 1.0 was released in April 2002, followed by Version 2.0 in September 2003, which is the current version.

AbiWord differentiates itself in its drive to become a fully cross-platform word processor and support for internationalization.

AbiWord has two native file formats .abw - an XML based file format - and .zabw which is a compressed .abw file. There is an AbiWord DTD, located at [xix] but this DTD is not used by the application to validate documents. Its purpose is rather descriptive rather than normative. It is not clear what the long-term plans for the DTD are [xx].

2.3.11 *DocBook*[xxi]

DocBook is general purpose XML and SGML document type particularly well suited to books and papers about computer hardware and software (though it is by no means limited to these applications).

The DocBook specifications are approved and published by the OASIS DocBook Technical Committee. DocBook is almost 13 years old. It began in 1991 as a joint project of HaL Computer Systems and O'Reilly. Its popularity grew, and eventually it spawned its own maintenance organization, the Davenport Group. In mid-1998, it became a Technical Committee (TC) of the Organization for the Advancement of Structured Information Standards (OASIS). The latest version of DocBook (V4.2.) can be found at [xxii];

DocBook's DTD defines more than 300 elements with an equal number of attributes. The DocBook markup language deals only with the semantic structure of the document. To work with DocBook the author must be able to work on the logical level of the document rather than on the visual level.

- In this context, WYSIWYG word processing is not possible in DocBook, because the way a section of text is to be marked up is not explicitly defined. Semantic markup makes the documents more amenable to interpretation by (publication) software however.

The author sees the document one way when it is created (e.g. with a plain text editor like notepad or emacs) and can automatically access completely different views through defined processes.

The philosophy of DocBook makes it more adapted to heavy structured documentation such as technical one, rather than general-purpose office documents. The great advantage that DocBook offers for technical documentation is that multiple publication formats can automatically be generated from a same DocBook source file:

- Printed version
- HTML
- RTF
- PDF
- Braille etc.

2.3.12 Adobe FrameMaker[xxiii]

Adobe FrameMaker 7.1 is a word processing tool that uses any XML DTD, as well as a binary format. FrameMaker gives users the ability to create, edit and import valid XML content in full WYSIWYG mode when authoring documents. Adobe FrameMaker supports DTD's but not the more recent XML Schema definitions.

The proprietary binary format, is recognizable by its .fm file extension. FrameMaker version 7.1 and version 7.0 are compatible, and can be opened on any platform. There is the option to save a document in FrameMaker 6.0 format for backwards compatibility.

Adobe has always kept FrameMaker as an upper-end product destined for technical writers and publishers. Its native format has therefore not been considered as a potential general office document format. Furthermore, Adobe's main interest lied in the end-form non-rewriteable, but high-fidelity PDF format.

2.3.13 Adobe PDF[xxiv]

Portable Document Format (PDF) is a file format developed by Adobe Systems for representing documents in a manner that is independent of the original application software, hardware, and operating system used to create those documents. A PDF file can describe documents containing any combination of text, graphics, and images in a device independent and resolution independent format. These documents can be one page or thousands of pages, very simple or extremely complex with a rich use of fonts, graphics, colour, and images.

PDF is primarily the combination of three technologies:

- A cut-down form of PostScript for generating the layout and graphics,
- A font-embedding/replacement system to allow fonts to travel with the documents
- A structured storage system to bundle these elements into a single file, with data compression where appropriate

PDF is a subset of those PostScript language elements that define the graphics, and only requires a very simple interpreter. For instance, flow control commands like *if* and *while* are removed, while graphics commands such as *lineto* remain.

There are several advantages to the PDF format. One is that there is only a single small file to transfer, whereas with the same file in PostScript format one must send the additional materials on its own. In addition it is faster to display on the screen. Finally, if displayed with Adobe's Acrobat Reader, there is a font-substitution strategy that ensures the document will be readable even if the end-user does not have the "proper" fonts installed.

When PDF first came out, in the early 1990s, it was slow to catch on. At the time, not only did the only PDF creation tools of the time (Acrobat) cost money, and so did the software to view and print PDF files. Additionally, there were competing formats. Adobe started distributing the Acrobat Reader program at no cost, and continued to support PDF through its slow multi-year ramp-up. Competing formats eventually died out, and PDF became a well-accepted standard.

Several independent PDF viewers and interfacing libraries have been developed, for example Xpdf, and GNOME Pdf for POSIX-like systems.

In recent versions, the functionalities of PDF have been extended. They can also include interactive elements such as buttons for forms entry and for triggering sound and Quicktime or AVI movies. PDF files are optimized for the Web by rendering text before graphic images and hypertext links.

Similar to Postscript, PDF remains an end-form format which is not intended for re-writing documents.

2.3.14 Adobe XDP[xxv xxvixxvii]

XDP (XML Data Package) format has recently been announced by Adobe.

An XDP file is simply an XML file that packages a PDF file in XML, along with XML form and template data. In this way, XDP represents both the logical structure of documents in XML as well as their presentation in PDF.

An XDP file contains several distinct blocks of information:

- **XML Form Data**. This component is the user data encoded according to an arbitrary XML schema chosen by the form developer during the design phase. The schema can be an industry standard, the enterprise's standard, or completely customized. Some examples of industry-standard schemas are ACORD (insurance), XBRL (finance), HL7 (healthcare), and SF424 (eGovernment).
- **XML Form Template**. This component contains all the form intelligence, including the mapping of XML form data to PDF form fields as well as all the business logic that controls the interactive behavior of the document, such as calculations and data validations.
- **XML Configuration Information**. The XML form template uses this component as a global reference for database and Web services SOAP connections.
- **Other XML Information**. XDP files can include custom XML information such as a schema file to facilitate validation, XML digital signatures, content metadata to facilitate archiving, or data used by a custom digital document application.
- **PDF Document**. XDP files provide all the traditional PDF benefits of precision document layout and high fidelity printing by embedding the PDF in an XML element (base64 encoded).

An XDP file is an XML file, so all XML tools, XML system interfaces, and Web services can work directly with it. The XML data is directly accessible. The new document format has been developed to provide organizations with a step-by-step migration path from manual, paper-based workflows to streamlined, automated processes that fully integrate electronic documents and forms.

There are different levels of PDF electronic forms sophistication:

- As a first step, simply converting paper-based forms to PDF reduces the distribution costs by enabling forms to be e-mailed or downloaded from a Web site. Users can then print a form, fill it in, and send it back.

- As a second step, users can fill in forms online and then print them. This step improves the legibility of the captured information and, when calculations and validations are included in the form, the accuracy of the data is improved.
- The third step is to enable electronic submission of the form, including applying digital signatures. The XML data can be directly integrated into back-end systems, thus reducing errors associated with rekeying.
- The final step enables forms to initiate and continue through complex business processes, with data being added to the form as it moves through the workflow and integrated into multiple systems along the way. Advanced network services, such as Web services or data connectivity, can also be addressed for more sophisticated business interactions.

The XDP specifications will be published and made publicly available.

XDP is very promising in that it adds to PDF the possibility to include the re-writeable XML counterpart. However, XDP seems to be more focussed on representing electronic forms. The re-writeable part is therefore a form-oriented XML schema, which allows end-users to fill-in electronic forms, rather than any general-purpose document.

2.4 Discussion

2.4.1 Interoperability and fidelity

A scenario that illustrates an ideal world where all of the ODF criteria are fully respected can be as follows: A person uses a tool T1 on hardware/software platform P1 to create a complex document D with graphics, structured text and incorporated spreadsheets, transmits it in the form of format F to another person who opens/modifies it using tool T2 on hardware/software platform P2 and sends it back. The initial creator should be able to open the document as if it were the original one that was sent, ie. exact layout, content and structure, with the modifications included as if they were applied by the original tool.

All those who have attempted this scenario know that it is an almost impossible challenge in today's environments, even when using tools from the same vendor such as Microsoft. An MS document may look – both on-screen as printed out - differently depending on the Word version, platform (operating system and version) and even local computer settings (installed fonts, video card driver, printer driver, etc.). Compatibility between different versions of Microsoft Word is limited to a backwards compatibility and implemented through import and export filters, which do not always work flawless [xxviii].

An recent presentation[xxix] from CNES (Centre National des Etudes Spatiales, France) pointed out that documents at CNES which were created using word processors in 1985, had to be re-keyed in manually in 1990 (MS Word 2), and then again in 1997 (MS Word 95). I.e. the upward compatibility chain was broken in less than 10 years. Moreover, documents created in 1995 had to be restructured manually in more recent Word versions, and all mathematical formulae re-inserted.

When documents are to be exchanged between tools from different vendors the challenge becomes even more complex, since the interpretation of the format is made by different companies who do not necessarily share all knowledge and views about the format, and because the functionalities represented in the format might not be provided by the tools.

More precisely, interoperability is influenced by three factors[xxx].

- The **quality of the filter component**, that is, the quality of the component that translates the content and structure of a document into the internal representation of the application. Documentation on the structure of the Microsoft Word file format for example is only available up to version 97. The newer versions of Word do not offer a separate filter for saving in Word 97, which leads to the assumption there are no essential evolutions in the format. Nevertheless, there are differences in file size observed. Word 2003 changes the size of the test files originally created in Word 2000 when they are opened and saved in the doc format[xxxi].

- The **compatibility of the feature set** of the applications that exchange a document.
- The behavior or **interpretation of features** that exist in both applications. Even if two support a certain feature, there might be differences in the way the feature is actually applied, so that documents in fact might look different although they have exactly the same values stored for the feature within their text engines.

One (already resolved) example of this is the upper paragraph margin that exists in Microsoft Word and StarOffice/OpenOffice.org Writer. The latter, like many DTP (Desk-Top Publishing) and professional word processing applications, ignores this value for the first paragraph of a page. Microsoft Word does not. The result of this is that the documents look different, although exactly the same values are stored in both text engines.

Adobe’s PDF has gone a long way in the preservation of document layout, but had to sacrifice the ability to modify the document. PDF provides on-screen look and print output that have the industry’s highest level of fidelity, but even then they are not 100% perfect [xxxii] [xxxiii]. The PDF encoding utility configuration can largely influence the size of the generated PDF file but also (in the opposite sense) the quality and portability of the document:

- Compression may degrade quality of the embedded images since it often applies algorithms that loose information.
- Embedding fonts in a document ensures that the fonts will be the ones you choose, even if those fonts are absent from the computer displaying the document. For instance, if a font is not embedded in a PDF file and the user does not have access to the original font on their system, Create Adobe PDF Online substitutes the font with a Multiple Master serif or sans serif typeface. Embedded fonts are disabled by default because they increase file size, but in this case fidelity is not guaranteed since restitution is based on different fonts from the original ones.

This brings us to the conclusion that currently no document format exists that guarantees absolute format fidelity. We could rank the currently existing document formats on their level of format fidelity as:

- **High:** With the exception of differences due to locally installed fonts and tool bugs, the document layout is preserved across platforms and computers.
- **Medium:** There are important issues associated with the document format. This can be due to architecture dependency (binary formats) or incomplete documentation of the format (Word versions)
- **Low:** the document layout depends on the user's viewing preferences (XHMTL).

For a format to truly preserve format across different word processors, platforms and computers the **format should be well documented** and all factors that might influence the representation should be **integrated** in the electronic document (internalise the external factors). For instance fonts should systematically be integrated and calculated fields should store not only the formula (e.g. =Today) but also the last calculated value. If the document is opened on a system that is unable to perform the recalculation, at least the last known value could be shown.

2.4.2 Formats vs. Criteria

Document Format	***Abbr***	Open	Non-binary	Modifiable	High-Fidelity	Crossp-latform	Common features	Advanced features	Wide-Adoption
WordProcessingML	WordML	Y	Y	Y	M	?	Y	Y	?
Word 2002	doc2002	N	N	Y	M	N	Y	N	Y
Word 2001	doc2001	N	N	Y	M	N	Y	N	Y
Word 2000	doc2000	N	N	Y	M	N	Y	N	Y
Word 97	doc97	N	N	Y	M	N	Y	N	Y
Word 95	doc95	N	N	Y	M	N	Y	N	Y
Word 6 for Windows	doc6	N	N	Y	M	N	Y	N	Y
Rich Text Format	rtf	?	Y	Y	L	Y	N	N	Y
WordPerfect9	wp9	N	N	Y	M	N	Y	N	Y

WordPerfect10	wp10	N	N	Y	M	N	Y	N	Y
(La)TeX	latex	Y	Y	Y	Y	Y	N	M	N
StarWriter 5.2	sw52	N	N	Y	M	N	N	N	N
OpenOffice.org 1.0	OOo1	Y	Y	Y	M	Y	Y	M	Y
Kword 1.2	kw1	Y	N	Y	M	N	N	N	N
AbiWord 2.0	aw2	Y	Y	Y	M	Y	N	N	N
FrameMaker 7.1	fm7	Y	N	Y	M	N	Y	N	N
Portable Document Format	pdf	Y	Y	N	Y	Y	Y	N	Y
XML Data Package	xdp	Y	Y	N	Y	Y	Y	N	Y
Postscript	ps	Y	Y	N	Y	Y	Y	N	Y
XHTML	html	Y	Y	Y	N	Y	N	N	Y
DocBook	docbook	Y	Y	Y	N	Y	Y	Y	N

The above table makes a difficult attempt to apply the listed criteria to the current document formats.

It is important to note that this table is not meant to select a winner format or tool for a given need. Rather, it is meant to assess the given formats against the ODF criteria, and select those that require further evaluation in this respect, especially from a market momentum perspective. We have therefore simply used Yes/No/Medium values for the criteria, with no further weighting.

As can be seen from the table, two document formats seem to be most compliant, namely:

- SUN/OASIS OpenOffice.org
- Microsoft XML Reference Schema's (WordProcessingML)

These two document formats seem to comply with almost all of the criteria, especially:

- Modifiable/re-writeable, non-binary
- Support for common and more advanced word processing features
- Medium to high-level fidelity

These two document formats will be further analysed in the subsequent chapters of this document, with regard to the remaining strategic criteria namely, openness, wide adoption and interoperability.

A few other formats meet some of the criteria, namely :

- Adobe XDP: Given the strategic importance in the context of this study for the format to be rewriteable, XDP will not be analysed further. It will be looked into only from the digital signature implementation perspective.
- (La)TeX : as we have explained in the previous sections, this format is only confined to the scientific community, which although is large, does not qualify as general wide adoption. Futhermore, we believe that now that more and more tools implement MathML, esp. OpenOffice, (La)TeX users will gradually migrate.
- RTF: The evolution of RTF followed closely the MS Word versions, and then stopped. We believe that RTF has been replaced by the MS XML Reference Schemas, which have been selected for further discussion.

3 The Microsoft XML Reference Schemas

3.1 Introduction

Microsoft Corp. has announced on November 17th the availability of Open and -Free License for the Office 2003 XML reference schemas [xxxiv]. Microsoft has first delivered the WordprocessingML schemas; the other schemas namely, SpreadSheetML and FormTemplateML, have followed on December 5th.

These schemas describe how information is stored when documents are saved as XML. By making these schemas available royalty free, Microsoft wishes to attain a "new level of transparency, interoperability, document portability and ease of communication".

The documentation seems complete which was not the case for the previous WordProcessingML versions. It will give Linux, Unix and Windows developers access to the word processing, spreadsheet and form template XML schemas for several Office 2003 applications including Word 2003, Excel 2003 and InfoPath 2003.

Microsoft claims that this will enable third party software companies to build products that seamlessly interoperate with Office 2003.

3.2 Technical aspects

3.2.1 WordProcessingML

All of WordProcessingML, SpreadSheetML, and FormTemplateML are defined as XML schemas.

A Word 2003 document can therefore be exported into XML, in a way that conforms and validates with respect to WordProcessingML Schema.

Microsoft Word 2000 and 2002 had already limited support for XML. Documents saved in HTML format in Word 2000 and 2002 have some embedded islands of XML data saved within them, but you could not use either Word 2000 or Word 2002 to natively create or save XML documents.

In addition to WordProcessingML, Microsoft Office Word 2003 also includes support for custom XML schema definitions[3] (XSDs), making it possible to attach one or more custom schemas to a given Word document. It allows the users to annotate the document with the elements found in the attached schemas. This makes it possible to inject business-related markup into documents so they can be processed around business markup instead of the more generic WordProcessingML markup. For example, a document that contains information over a new employee could be annotated with elements of the employee schema. Via the template and add-ins dialog, it is possible to manage the schema library and to choose the schemas to attach to a particular document. The user can indicate if Word should validate documents and whether it should be possible to save invalid documents.

3.2.2 Main features of Microsoft Office 2003

The newest edition of Office focuses largely on collaboration: helping employees work with each other more effectively, whether they are sharing documents or planning meetings and events. Microsoft has mostly focused on making the existing tools easier to use rather than adding a large number of functionalities. [xxxv]

The main features added in the Office 2003 version are the following.

In Microsoft Word, users can:

- Lock down portions of Word documents to prevent editing.
- Assign permissions, enabling only certain users to make changes (the "compare and merge" feature is greatly improved, making it easier to see all reviewers' comments)

Document Workspaces allow users to:

- Co-author, modify and review files via a centralized, Web-based repository of files, tasks and lists of links and team members

[3] Schemas must adhere to the World Wide Web Consortium (W3C) XML Schema recommendation.

- Share attachments and contacts, giving their cohorts access to the latest versions of documents at any given time.

Microsoft Outlook 2003 has:

- New spam filtering capabilities
- An interface easier to use
- An interface that adds the ability to view multiple calendars simultaneously

Microsoft Excel 2003:

- 40 functions were rewritten that are useful for scientists and engineers
- MSExcel 2003 can use XML data
- Users can scroll through two spreadsheets, side-by-side, simultaneously

A new tool is now available either with the Office 2003 suite or purchased seperately : Microsoft Office OneNote 2003

- It allows users to organize notes in a single location.

Also, using the XML schemas, it is possible to integrate Office documents into business processes. For example, a letter is written to a client that includes the customer number, the subject and the date, each of these has a style attached to it. Based on the style, a down stream process can extract these fields and put them in a CRM system with a pointer to the actual document. The administrator just writes the letter and the rest happens automatically [xxxvi].

3.2.3 *XML features of Word 2003*

Word has two level support for XML. One could either simply use Word freely and save the document in XML which would generate a WordProcessingML compatible document. Or, define one's own XML schema and use Word as an XML editor similar to those available on the market such as Epic from Arbortext or Xmetal from Corel.

After a schema has been attached to a document, it is possible to start annotating the document with the elements of the schema.

The XML Structure pane also gives you an interface to the custom elements currently found throughout the document by displaying the logical tree structure of these elements within the pane.

There are two options for saving a Word document with custom elements. The default option is to save the document as WordProcessingML with the custom elements nested throughout the tree.

The other option is "Save data only", which removes WordProcessingML markup and only persists the custom element tree structure.

When the option you "Save data only" is chosen, Word removes the WordProcessingML markup and only saves the custom elements found in the document. Doing this, however, causes one to lose any special formatting that may have been applied to the document.

Word makes it possible to apply an XSLT transformation during the save process. One can use XSLT transformations to move between WordProcessingML and other text-based formats. For example to transform WordProcessingML into a HTML structure.

Word provides an Open as XML option in the Open dialog. When this option is selected, Word opens the result of applying a transformation on the XML document instead of the original document itself.

Features such as custom-defined XML schemas are offered only with for Professional and Enterprise editions.

3.2.4 Microsoft Office InfoPath™ 2003

New with the Microsoft Office System, InfoPath 2003 uses forms to let users enter information according to a customer-defined XML schema. InfoPath enables customers to gather and reuse information with predefined structure (pre-tagging) and as part of a business process [xxxvii].

InfoPath provides all the functionality expected from a forms package, including the ability to structure and validate data, as well as the ease of use of word processing—all within the familiar Office user interface.

InfoPath supports complex forms with hierarchical structures, freeform text, tables, optional or repeated blocks, data validation, data aggregation, and forms with need of multiple views. In a corporate environment, InfoPath streamlines data entry and data capture; native support for XML enables companies to create InfoPath solutions that send data from the desktop environment to backend systems via XML Web services.

The presence of InfoPath underlines Microsoft strategy towards “all XML”. This also underlines the importance for WordProcessingML to allow user-defined schemas.

3.3 Microsoft XML strategy

What makes XML unique is that it can represent structured, unstructured and semi-structured information. With XML there remain no barriers between document processing, content management, ERP and other systems. With XML we shift away from the paper-based document paradigm to one where documents and data are intertwined.

Microsoft has clearly gone deeply in this direction, as one can attest from the following observations:

Microsoft has been a very active member of the W3C. They followed on the XML evolution at every step.

The new architecture of Microsoft is entirely based on XML Web Services (WS) as provided by the W3C recommendations (SOAP, WSDL etc.). The major milestone will be Windows 2003 which will have Web Services semantics built in.

Microsoft has made 3 co-announcements with IBM in September regarding future improvements to Web Services, namely :

- *WS security* : Exchange of security credentials across security domains via a security broker.
- *WS reliability* endorsed by IBM and MS, and
- *WS transaction* or 2 way commit

All this means that WS barriers have fallen down and this will be part of Windows 2003. MS decided to accelerate this platform into the market place.

MS took the strategic decision of making office system XML compliant. Support for XML is full fledged:

- *No limit to one schema;*
- *Support for custom design schemas;*

MS does not believe that the market can be satisfied by a unique schema. MS decided also to have XML tagging for their internal format.

What MS did not do is to submit the XML Reference Schemas to a standardisation body. The first reason is that MS is not yet ready to give up intellectual property rights although royalty free. Second, documents will not exist anymore; MS believes we are at the beginning of the XML revolution where we will no longer be paper-based paradigm.

MS does not believe the patent licence will hinder vendors such as Oracle or IBM or other players from developing WordProcessingML compatible tools.

MS believes that standardising the XML Reference Schemas will bring the risk of slowing down their evolution, which could hinder development of future features in MS Office.

According to Microsoft, " the XML Reference Schema announcement underlines Microsoft's commitment to constructive dialogue with governments and the industry with regard to intellectual property issues. Microsoft listened to requests for clarification of its licensing policy with regard to the Office 2003 XML Reference Schemas and is now responding to those requests by delivering a world-wide open and royalty-free licensing program. Individuals and organisations, including governments, academics and commercial software vendors, can enter into the license.

This announcement builds on prior Microsoft efforts to promote interoperability, including development and standardisation work for XML itself, SOAP, UDDI, WS-Security (the standard security model for XML web services) and other XML industry standards. This licensing program is intended to complement these efforts.

XML is widely recognised as the next-generation technology for integrating applications, services and data sources. By providing an open and royalty-free licensing program for the Office 2003 XML Reference Schemas, Microsoft is helping to facilitate the free flow of data between disparate islands of technology.

Finally, by offering this license, Microsoft re-emphasises its commitment to make the Microsoft Office System a first-class development platform for XML. Microsoft recognises that XML web services can dramatically reduce IT integration costs while also improving the productivity of end users. By providing this new licensing program, Microsoft hopes to further underline its commitment to taking positive and constructive steps toward helping customers realise the full potential of XML”.

3.4 Main assessment criteria

3.4.1 Cross-platform Interoperability

By looking into the MS XML Reference schemas as a format only, and given the fact that they are expressed in W3C-compliant XML, one could in theory claim that they are portable to various heterogeneous platforms. However, as we discussed in section 2.1.4 on XML, this theory does not always hold true.

Recall that cross-platform interoperability implies that the format can be exploited, with full preservation of its semantics on various hardware (PC, SUN, Mac..) and software platforms (Windows versions, Linux and Unix versions under different desktop environments etc..).

Regarding MS XML Reference schemas and the above definition, we could express the following reservations :

ü MS XML Reference schemas elements can contain proprietary objects. These objects are encoded in a standard-compliant fashion (ex: base64, UTF-8, etc.) but some of them may be executed only in a Microsoft environment (ex: OLE).

ü The spreadsheet macros are spread within the content XML elements. It is therefore very difficult to isolate the code from the text by a third-party program. Furthermore, these macros cannot be executed outside the MS-Office environment.

3.4.2 *Wide adoption*

Given the recent announcement of the schemas, we could only make predictions regarding their future adoption:

Response to a user demand: Microsoft has made the schemas available publicly following increasing pressure from customers, partners, governments and the IT industry. As information exchange and integration have increased in critical importance, they have asked Microsoft to deliver solutions that improve data interoperability and exchange.

Discussions with the Danish government seem to have played an important role in this announcement. Microsoft wanted to respond to strong requests from governments, and namely the Danish one for interoperability and openness.

Success of Office 2003: The future of the XML Schemas cannot be dissociated from that of Office 2003 which will depend on user perceived added value of the product as well as the pricing policy.

The catalogue prices (new user prices are around 500$) can be a hindrance to the uptake of MS Office in certain developing countries administrations where Microsoft is confronted with severe Open Software competition. However, prices may vary widely. The press speaks of as down as 20$ discount rate offered to Thailand Administration.

3.4.3 *Openness*

The minimum requirements for an open standard are that the document format is completely described in publicly accessible documents, that this description may be distributed freely and that the document format may be implemented in programs without restrictions, i.e. royalty-free, with no legal bindings.

The MS license provides access to the schemas and full documentation to interested parties and is designed for ease of use and adoption. In this regard the MS XML Reference schemas satisfy the requirements.

The associated legal terms seem to create a lot of controversy. As we are not qualified to make a judgement on this basis, we will simply highlight the main points hereafter, and recommend examining carefully the legal aspects of the licence.

3.4.3.1 The intellectual property

The intellectual property remains with Microsoft. The license precludes the modification or extension of the schemas. Microsoft is not offering these schemas to a standards body.

Patents are associated to the license.

[Patent License excerpts:] "Microsoft may have patents and/or patent applications that are necessary for you to license in order to make, sell, or distribute software programs that read or write files that comply with the Microsoft specifications for the Office Schemas..." [xxxviii].

"Except as provided below, Microsoft hereby grants you a royalty-free license under Microsoft's Necessary Claims to make, use, sell, offer to sell, import, and otherwise distribute Licensed Implementations solely for the purpose of reading and writing files that comply with the Microsoft specifications for the Office Schemas.

The schema download contains language that allows to copy and distribute the schema, subject to certain limitations (credit it and link to a particular page at Microsoft). But the download doesn't grant the right to implement a program that can use the specifications [xxxix].

This part is ambiguous. Two theories conflict on this manner.

The first one translates "*not being licensed to distribute under other license terms in the Patent License*" as a clause designed to prevent application that use the Gnu General Public License (GPL) from implementing Office XML compatibility. Developers writing open source software should be careful before using these schemas [xl].

The second, more positive, is from Eben Moglen, the Free Software Foundation FSF's pro bono counsel. He told www.theregister.co.uk. he didn't think " *the alarm is justified." "This is not a license that I would like to accept; Microsoft is saying we might have some patents. But it's not a problem if Microsoft is making it available to everyone to make use and sell.*"

While Microsoft will make available the Office schemas, the company will retain control over how those schemas are developed in the future. That puts the burden on competitors to keep up with Microsoft's changes.

Microsoft is committed to making updates to the Office 2003 XML Reference Schemas available under the same terms and conditions as the licenses offered on November 17, 2003. MS will use the same royalty-free licensing terms for any updates, revisions or edits to the Office 2003 XML reference schemas [xli].

At the same time, Microsoft reserves the right to change its policy and/or the terms of the licenses with respect to future versions of Office [xlii].

3.4.3.2 *Rights to modify/update*

“You can distribute your program in source code form, but note that the patent and copyright provisions in the license for the Office 2003 XML Reference Schemas require you to include a notice of attribution in your program”.

The actual Microsoft patent statement says you must obtain a license if you use the information in a separate application for compatibility.

Quoting them:

"There is a separate patent license available to parties interested in implementing software programs that can read and write files that conform to the Specification." [xliii]

3.4.4 *Compatibility with OpenOffice.org*

Microsoft Office 2003 does not offer an import/export filter to OOo format. OpenOffice/Staroffice provide filters to earlier versions of MS Office. The development of such a filter is purely a political/strategic decision, since they are technically feasible although with fidelity loss in some cases.

Microsoft has made no announcement as to any future development of such filters. Analysts believe that Microsoft will never develop import/export filters to OpenOffice.org, as this would undermine their position as the de facto reference document-processing tool.

In theory, Word 2003 shoud be able to edit OOo documents, if one considers the OOo schemas as “user defined”. We could not at this stage make a pragmatic opinion on this point. On one hand, Word2003 and WordProcessingML are very recent. On the other hand, making OOo work as a specific user defined schema implies developing all the XSL transformations, which could prove extremely difficult to manage for a large and heavy schema as OOo. Furthermore, this task could prove as complex as developing an entire word processor from the start.

4 OpenOffice.org

4.1 Introduction

4.1.1 OpenOffice.org

OpenOffice.org (OOo) is an office applications suite. OpenOffice.org is a community-based project and is based on the open-sourced code from an older version of StarOffice created by Sun Microsystems.

The goal of the OpenOffice.org community is to "create the leading international office suite that will run on all major platforms and provide access to all functionality and data through open-component based APIs and an XML-based file format." As described in the overview document, "OpenOffice.org is both an Open Source product and a project.

The product is a multi-platform office productivity suite. It includes the key desktop applications, such as a word processor, spreadsheet, presentation manager, and drawing program, with a user interface and feature set similar to other office suites. OpenOffice.org also works transparently with a variety of file formats, including those of Microsoft Office.

OpenOffice.org is available for download on the OpenOffice.org website and distributed by partner vendors.

4.1.2 History

Staroffice is 17 years old. In August of 1999 Sun Microsystems purchased StarDivision, a German software company who produced an office suite known as StarOffice and which was founded in 1991. Sun's strategy at the time was to provide an alternative office suite to the dominant Microsoft Office.

In 2000, Sun released the source code of StarOffice software publicly through OpenOffice.org, thus initiating the world's largest open source project. The OpenOffice.org community includes over 14,000 developers and 100,000 registered members, working on more than 45 projects and responsible for the ongoing development of

OpenOffice.org. More than 40 million copies of OpenOffice.org and StarOffice software have been distributed to date.

OpenOffice.org uses a dual-licensing scheme for source-code contributions: the LGPL (GNU Lesser General Public License) and SISSL (Sun Industry Standards Source License). For documentation and website content not intended to be included in the product, the Public Documentation License (PDL) is used.

The OpenOffice.org source code initially includes the technology that Sun Microsystems has been developing for the future versions of StarOffice(TM) software. The source is written in C++ and delivers language-neutral and scriptable functionality, including Java(TM) APIs. This source technology introduces the next-stage architecture, allowing use of the suite as separate applications or as embedded components in other applications. Numerous other features are also present including XML-based file formats and other resources [xliv].

4.2 SUN OpenOffice strategy

SUN's strategy is mainly driven by their increasing awareness, especially after the Internet bubble burst, that the future of UNIX systems cannot be limited to the server side.

SUN has therefore deployed all its effort to provide JAVA based products and solutions to sit on all devices including PDA and desktops.

UNIX could not become a desktop platform without a productivity office suite. This, coupled to SUN's long-term culture for openness, gave rise to SUN's decision to both acquire StarOffice and put it in the Open Source realm, whilst maintaining the development and sale of few add-on's in order to keep revenues.

SUN is committed to OpenOffice.org in the same way it is committed to JAVA. The driving rationale is that this way market size will grow, together with Sun's share, at the expense of monoculture products.

OpenOffice.org Software Support is available on the Sun Microsystems web site [xlv]. It includes various level of support from standard to round the clock 24hr support. Sun also provides migration services for users who plan to migrate from another office productivity software to OpenOffice.org.

4.2.1 Differences between StarOffice and OpenOffice.org

SUN's StarOffice software is based on the OpenOffice.org suite and includes value-added enterprise capabilities and features.

The source code available at OpenOffice.org does not consist of all of the StarOffice code. Usually, the reason for this is that Sun pays to license third party code to include in StarOffice that does not have permission to make available in OpenOffice.org [xlvi].

StarOffice 7 is a snapshot of the OpenOffice.org 1.1 development tree. Some bits of proprietary code remain in StarOffice 7, such as the spell checker, which gives different suggestions than the OpenOffice.org spell checker. However, the main differences are the enhancements bundled with StarOffice, which include:

- A different, more colourful set of icons, both for task bars and for StarOffice-associated files in a file manager
- Nine proprietary fonts, ranging from the workhorse fonts Arial Narrow and Garamond to the less commonly used decorative fonts Palace Script and Broadway
- Sixty-day setup support
- A 482-page manual covering basic features and including some tutorials
- An extensive set of templates for Writer and Impress

Functionalities that are or will be present in StarOffice but are not available on OpenOffice.org:

- Fonts (including, especially, Asian language fonts)
- Database component (Adabas D)
- Templates included with StarOffice
- Extensive Clip Art Gallery
- Sorting functionality (Asian versions)
- File filters

Detailed comparison of both products can be found at [xlvii].

4.2.2 *Pricing policy*

As mentioned earlier, SUN sells and provides support for both of OOo and SO, according to the following general terms :

- Ooo 1.1 : free, SDK, 24x7 support, multiple platforms
- SO 7 retail: 80$ MSRP, enhanced spellchecker, fonts (since Linux does not have them), clip art, 60 day support entitlement
- SO7 Enterprise (target market): 60$/user, premium support, configuration manager, migration tools, enterprise guides, partner programs + migration tools (in pilot for the moment)

SUN's current main focus is on migration tools, since migration costs are the main barrier for SO adoption.

4.3 Main assessment criteria

4.3.1 *Openness*

The OpenOffice.org format is an XML format, which is fully documented and freely available from the OpenOffice.org open source community. Its use and extensibility is provided freely with no legal constraints.

Furthermore, OpenOffice format is being standardized by OASIS, the "Organization for the Advancement of Structured Information Standards" [xlviii].

OASIS is a not-for-profit, global consortium that drives the development, convergence and adoption of e-business standards. OASIS has more than 600 corporate and individual members in 100 countries around the world. OASIS and the United Nations jointly sponsor *ebXML*, a global framework for e-business data exchange.

The purpose of the OASIS OpenOffice Technical Committee is to create an open, XML-based file format specification for office applications. The resulting file format must meet the following requirements:

- It must be suitable for office documents containing text, spreadsheets, charts, and graphical documents,

- It must be compatible with the W3C Extensible Markup Language (XML) v1.0 and W3C Namespaces in XML v1.0 specifications,
- It must retain high-level information suitable for editing the document,
- It must be friendly to transformations using XSLT or similar XML-based languages or tools,
- It should keep the document's content and layout information separate such that they can be processed independently of each other, and
- It should 'borrow' from similar, existing standards wherever possible and permitted.

4.3.2 Cross-platform interoperability

This criterion holds true for the OOo format, at least for the platforms on which the OpenOffice/StarOffice tools have been implemented.

Nothing in the OOo format as such should prevent it from being processed on further existing platforms, or future ones.

4.3.3 Wide Adoption

4.3.3.1 User momentum

OpenOffice.org spokesman Sam Heiser predicts that OpenOffice.org will become the dominant desktop productivity standard within the next 10 years [xlix]. The project is especially trying to drive home its message to small business users, especially if the state of the economy continues to add pressure to contain costs.

The user base is difficult to establish precisely, mainly due to the fact that one could not distinguish users from those who simply download. One thing seems to be sure, however, is that OOo adoption in governments and Administrations is beyond any doubt as the following sample illustrate [l]:

France : November 2003: the French Ministry of interior has already migrated 15,000 desktops to OpenOffice.org and intends to migrate another 15,000 on the 100,000 ministry desktops within a year.

United Kingdom: December 2003:British government contract to fit 500,000 computers with Staroffice.

Germany: The German government is moving to standardize on Linux and an open-source IT model at the federal, state and communal levels. The city of Munich, the third largest in Germany, has also chosen Linux and the free OpenOffice.org productivity suite for its more than 15,000 desktop systems.

Other countries and administrations include China, Thailand, Israel, Australia, Philippines, Uganda, and Vietnam. The main criteria in favour of OpenOffice/Staroffice adoption are the price (free or extremely low), openness, and multiplaform capabilities especially for Linux and Windows.

4.3.3.2 *Vendor momentum*

Vendor momentum is undoubtedly much inferior than user momentum. This could be explained as being due to the following factors:

OpenOffice has not yet been voted by Oasis. As such, vendors are not inclined to commit to it at this moment.

The user base has grown only recently. Vendors have had no time to adjust their strategies and R&D roadmap.

Vendors are not willing to make any public announcements before their software products are out, lest they upset Microsoft.

As of today, public announcements have come from:

Koffice, an Open Source initiative, (individual) member of Oasis OO TC. Koffice will make of OpenOffice its native format for the future versions.

SoftwareAG, who announced in March 2003 connectivity between their XML database Tamino and StarOffice. This not yet part of the product catalog, but the product is enabled already for StarOffice and available for Tamino developer community. OpenOffice is not supported.

Other vendors include:

SAP, who is developing interfaces to Staroffice

Sage: Sage's strategy is to supply the technology that their customers want, mainly small businesses who are not early adopters. Sage is watching with interest the evolution of OO market in order to provide OO compatibility as soon as the market opportunity arises.

4.3.3.3 Standardisation momentum

There are 7 levels of membership at OASIS ranging from simple observer, to members with voting rights. Voting members at the OASIS technical committee are as follows:

Arbortext, Corel Corp., Koffice (represented via individual membership), National archives of Australia, Society of Biblical Literature, SUN Microsystems and Boeing.

SUN and Koffice seem to be the most active members, plus Corel to a lesser extent.

Voting members have to attend 2 out of 3 meetings. For many this is too high a constraint, which means that they opt for the observer membership. Unfortunately this is not a public list.

OASIS uses OOo format as a base and apply some minor technical changes. These changes have all so far been integrated by SUN and OpenOffice.org.

According to the OO Technical Committee chairman, Michael Brauer from SUN, the format will be out for vote around Q1-Q2 2004. He believes that the format should be voted as a standard in first semester of 2004. Technical discussions are still on regarding which of DTD, Schemas or RelaxNG will be used as the normative reference description language of the Oasis OO format.

Once OpenOffice.org is a standardized format, companies, governments and users in general will be more willing to use this format. Users' acceptance will also allow software editors to develop new programs in open source.

It is noteworthy that Oasis OO TC members Corel and Arbortext have so far made no public announcements regarding any support of OO.

5 Digital Signature

5.1 Digital signature concepts

5.1.1 Public key encryption

Public key encryption, also called "Asymmetric encryption" is based on the usage of a pair of keys: a private key, which has to be kept private by its owner; and a public key that can be made available to the general public.

Either key can be used to encrypt a message, but only the opposite key in the pair can decrypt the message. For example, if a message has been encrypted with the public key, then only the private key can decrypt the message.

It can be used for:

- When someone signs information with his private key: it helps recipients authenticating of the sender/author; this is used for digital sign information;
- When someone signs information with someone's public key: he can be sure that only the owner of the corresponding private key can read it; in particular this can be used when sending confidential information through email.

Public key encryption's advantage is the management of keys, as private keys need not be distributed to the recipient. On the other hand, one of the major drawbacks is that asymmetric encryption can be very time-consuming when applied on large files.

5.1.2 Digital signature

A digital signature, which is nothing more than an electronic signature, has two primary purposes:

- *Authentication*: to authenticate the identity of the sender of the message;
- *Data Integrity*: to ensure that the original message has not been changed (since the data were signed).

In other words, a digital signature proves that the owner of a private key is the author of the signature and that the data has not changed since the signature.

Note that the purpose of a digital signature is NOT to ensure confidentiality.

As the process of asymmetric encryption is time-consuming, the digital signing of a document relies on the asymmetric encryption of a digital fingerprint (usually called digest), much smaller in size, of the document.

The digest is obtained by a hash function on the document. A hash function is a transformation that takes a variable size document and generates a corresponding fixed-size string (the digest) and for which the chance that a different document has the same digest is very low.

Therefore, digitally signing is a two-phase process:

- Calculation of a digest of the document;
- Asymmetric encryption of the digest with the author/sender's private key.

There are two different cases to distinguish:

- The signature is embedded in the document; the signed document is self-contained;
- The signature is held in a separate file.

5.1.3 Digital signature of XML documents

The W3C has developed a standard (W3C XML Signature) to normalise the representation of digital signatures in XML. As an open standard and as it is format-independent and rely on XML, it ensures portability and openness of digital signature. W3C XML Signature standard has been adopted by major industry players as the format to store and/or transport digital signatures.

W3C XML Signature specifies how to carry multiple digital signatures in XML. Those signatures can refer to any content whether they are XML or not.

XML signatures can be *enveloping* i.e. a signature over content within the same XML document (this allows generating self-contained signed XML documents) or *detached* (signatures over external data).

5.2 Digital signature features

5.2.1 *Digital signature capabilities*

Digital signing data held in a file can be achieved independently of the software that was used to create it by using a third party tool that will treat the content of that file as an octet stream. As a consequence, any file can be digitally signed (producing a detached signature contained in a file) whatever format it is stored in and whatever application was used to create it.

Nevertheless, in order to efficiently use digital signatures on document, the functions must be easily available to end-users.

As regards digital signatures, the following functions covered by Office Document applications and formats will be assessed:

Function	**Description**
Signature	Can you sign a document from the Office software?
Validate signature	Can you validate the signatures on a document from the Office application? Can you validate certificate through a PKI/RCL server?
Multiple signature	Can several people sign the same document through the Office software?
XML digital signature compliant	Does digital signatures comply with the W3C XML Signature standard?
Enveloping/Detached signatures	Are digital signatures included in the document itself of are they stored in an independent file?
Partial signature	Can you digitally sign only a specific part of the document?

5.2.2 *Adobe PDF/XDP*

Function	**PDF**	**XDP**
Signature	Yes[4]. Adobe Acrobat 6.0 allows users to digitally sign any acrobat file.	
Validate signature	Yes.	
Multiple signature	Yes. The same file can be digitally signed by several different users.	
XML digital signature compliant	No. In this case the digital signatures are stored in the PDF file itself in a format specific to Adobe Acrobat.	Yes. A specific section of the XML XDP file contains the digital signature(s).
Enveloping/Detached signatures	Enveloping. In PDF, the signature is held in the PDF document.	Enveloping. The XDP format is an XML file that “packages” different pieces of information (form data, form template, PDF files, etc.). Those different pieces of information are held in different blocks (i.e. XML element with different namespaces) of the XDP file, one of which being dedicated to digital signatures.
Partial signature	No. When digitally signing a document, a user can specify whether he is signing the entire content of the document or only a part of it. Nevertheless, in the latter case, the user cannot specify explicitly the part(s) of the document that he certifies or agrees with by signing the document.	

[4] Signing cannot be achieved using Adobe Acrobat Reader. It requires the use of Adobe Acrobat Standard or Professional.

5.2.3 OpenOffice

There are no signature features available in the current version of OpenOffice (1.1).

Nevertheless, it is planned to introduce digital signature capabilities in the next (also called the "Q" release) version, which should be released by the end of 2004.

The following table describes the functions that should be available in the Q release as stated in the product concept [li] document available on the OpenOffice website:

Function	**Open Office**
Signature	Yes.
Validate signature	Not clearly stated in the document.
Multiple signature	Yes.
W3C XML digital signature compliant	Yes.
Enveloping/Detached signatures	Although this is not explicitly said in the document, it is likely that the signatures will be added in the ZIP archive as separate XML documents (W3C XML signature compliant).
Partial signature	Not clearly stated in the document.

5.2.4 *Microsoft Office 2003*

Function	**Microsoft Office**	**InfoPath**
Signature	Yes. With Microsoft office 2003 (and XP) applications, a user can easily sign a document. There also exist third-party solutions working with previous versions that can add basic digital signature functions. *Note that it is not yet clear whether the digital signature is exported when saving a file as XML.*	Yes.
Validate signature	Yes. A digital signature can be validated.	Yes.
Multiple signature	No. Only one user can digitally sign a document.	Yes. Several users can add their own digital signature to the same document.
W3C XML digital signature compliant	No.	Yes.
Enveloping/Detached signatures	The digital signature is included in Microsoft Office documents.	The digital signatures are included in the XML document produced by InfoPath (conformant with the W3C XML signature format).
Partial signature	No.	No. This function is planned for the next release (mid-2004).

6 Conclusion

6.1 Market share

Software editors share the Office suite market quite unfairly. Microsoft is the dominant vendor with an approximate 95% market share (according to the Giga Information Group). Its competitors share the 5% market share left.

Market share: Worldwide Office suites installed base

Office suite	Market share end 2002
Office 2000	50%
Office 97	33%
Office XP	11%
WordPerfect (Corel)	2%
SmartSuite (Lotus/IBM)	2%
Other (including StarOffice)	2%

(Source: Giga Information Group, estimated desktops: 250 millions - http://www.journaldunet.com)

Although the acceptance and adoption of Open Source desktop suites is increasing, Microsoft's dominant position doesn't seem threatened yet by another vendor but himself [lii]. A Giga pole shows that Microsoft clients wonder more often whether they will upgrade to existing Microsoft suites more than migrate to an alternative one.

Office Suite Adoption Plans

Plan	Percent of Responses
Evaluating upgrade to existing MS suite	42%
Evaluating migration to alternative suite	11%
Not performing an evaluation at this time	45%
No answer	2%

Source: Giga Information Group)

Only 42 % are positive about a migration to an existing suite. 11% of the clients surveyed say they are evaluating migration to an alternative suite likely leading to a market share loss in the following 2 years.

6.2 What will impact the adoption of OpenOffice vs. MicroSoft?

What might make users choose an alternative path to Microsoft? Various criteria, such as organizational, functional and financial, could impact the market. The principal ones are detailed in this paragraph.

6.2.1 Linux OS adoption

The adoption of Linux is increasing worldwide, mostly in developing countries. This will lead to an equivalent increase of Linux desktop adoption and open source, multi-platform office suites.

This increase will be restrained by the small range of peripheral devices and the difficulty for desktop users to install and get access to critical ones. A significant amount of work remains to be done before peripheral device support on Linux will be acceptable for a mainstream consumer.

6.2.2 OpenOffice functionality and compatibility with windows

According to various sources, OpenOffice is now “good enough” for significant classes of users”, and StarOffice adds more and more functionality.

End 2002, one-third of all desktops was still using Office 97. The end of the Office 97 support is programmed for soon. Will companies and governments migrate to office 2003 or XP or will they consider an alternative suite? According to Gartner, Office 2003 new features might be not attractive enough for enterprises to consider. [liii]

According to the Gartner Group [liv], StarOffice functionalities are comparable to the functionalities of Office 97. Governments, companies and individuals considering that these are enough for them may not follow the Microsoft path.

Although Microsoft has released the XML reference schemas for the Office 2003 version, these schemas aren't for individual or everyday use. They are intended to an experienced crowd and for specific use.

6.2.3 *Microsoft new licensing policy*

Microsoft may have little choice but to resort to increasingly aggressive upgrade strategies and (such as its new licensing practices) for the desktop operating system productivity suite that bring in the bulk of its revenues, encouraging customers to look for cheaper replacements for commoditized software.

According to the Gartner Group [lv], Microsoft has upset many customers with the introduction of Licensing 6.0. Michael Silver, a Gartner analyst, said some firms are beginning to weigh the cost and licensing terms of Microsoft's Office against StarOffice's improving compatibility with Microsoft file formats and its expected lower pricing. "StarOffice has a chance, based on better compatibility, some mind share and Microsoft missteps" [lvi].

6.2.4 *Costs and savings*

"Vendors operating on razor-thin margins and price-conscious customers will naturally look to save on the most expensive components." [lvii]. IT managers facing a long and deep economic slump are also under pressure to rationalize their computer infrastructure.

According to Soreon Research [lviii], large businesses can save up to 20% by using the OpenOffice.org open source office application instead of Microsoft Office. If using Linux instead of Windows, it is possible to save up to 30%. On the other hand, savings for smaller businesses get all insignificant as the size of companies decrease.

The Soreon model shows that especially big companies can save big with Open-Source. Over a 3-year-period large enterprises with 2000 workstations using Open-Source can save 525,000 € through office-products, 57,000 €, through servers, 32.000 € through content management systems and 21.000 € with Open-Source databases.

6.2.5 *Deployment*

The major constraint is deployment cost. Migration, training and resistance are the main reasons a large number of companies are still hesitating to take the plunge. " Migration costs, end-user training and converting documents from Microsoft file formats could deter companies" [lix]. Decision makers are awaiting success stories to go further in their decision process.

6.2.6 *Look and feel of the applications*

Despite the improvements carried on the appearance of Openoffice.org and the smoother look and feel of the application, Microsoft remains ahead in the "design" race.

Analysts agree erosion is ahead of Microsoft's office suite market share. This will not impact its dominant position. The combination of these criteria might lead to 5% erosion in the following 2 years.

6.3 Where is the market heading?

Many figures have been out before the release of Microsoft's XML Reference Schema.

The following articles were released before the recent announcement of Microsoft's XML reference schema. Some analysts have commented the impact of this announcement as said in the above chapters of this document. We expect in the upcoming weeks detailed analysis regarding this disruptive change by major market analysts.

They highlight the major indications:

- According to French 01net.com (*October 2003*), 20% French companies intend to migrate to Office 2003 but the Open Source is becoming threatening. Companies won't migrate before 6 to 12 month [lx] .
- According to Gartner (*April 2003*), the small number of companies willing to switch to Office 2003 demonstrates that this new version isn't convincing them. It also demonstrates that the end of the Office 97 support is not a good enough reason to upgrade. [lxi]
- According to the Giga Information Group (*2003*): "Within the next two years, the arrival of attractively priced competing office suites combined with dissatisfaction with current Microsoft licensing plans will create upwards of a <u>5 percent market share loss</u>, (…)". This loss will have a strong impact

on Microsoft’s benefits since Office turnover represents more than 40% of Microsoft’s total benefits.

- According to xmlcoverpages.com (*December 2003*): Way behind, Openoffice.org has had almost 19 million downloads from its official sites [lxii]. “Since the software can be given away freely, this total could mean that some 60-80 million copies are floating around. In comparison, the unofficial word is that StarOffice has sold some 50 million licences - and that was before Sun's recently announced deal to supply the Republic of China.” These figures are based on the number of desktops worldwide estimated around 650 million according to sales during the last 5 years (*IDC*).
- In May 2002 Gartner is also estimating that “StarOffice has a slightly better than 50% chance of taking 10% of the office productivity suite market - at Microsoft's expense - by the end of 2004 (*May 2002*) [lxiii].

Microsoft has announced the publication of the XML Reference Schema. Will these schemas attract the desired population? Will Microsoft trigger a change with this openness attempt? Governments and companies that have already invested in Open Source might not change their policies, but will the others do?

In its attempt to standardize the OpenOffice.org format, Sun Microsystems is not backed up by other market players except for the Open Source community. It is also likely that this standardization process will not have the same impact as it should have if Microsoft hadn’t released its XML schema.

Now that the market is not Microsoft vs Opensource or Proprietery vs Openness since the publication of the schemas, two ‘standards’ will be fighting their way towards public acceptance. This of course presupposes that the legal issues concerning Microsoft licences of the schemas have been clarified.

In any case, our view is that the corresponding formats, namely MS XML Reference schemas and OOo will naturally follow the adoption of the major tools behind them. Microsoft’s XML lead and market dominance will remain for the few years to come. On the other hand, OpenOffice user base size is now such that it is irreversible, and it constitutes a viable alternative to Microsoft. In terms of wide adoption of the format, we believe that none of the two will be winner or a knockout looser, with MS dominating the user base at 85%. The two formats will coexist, but OOo will become more and more the open format reference for interoperability across platforms.

7 References

[i] BSI Report

[ii] ZapThink report on XML. http://www.zapthink.com

[iii] *http://www.nhsia.nhs.uk/napps/step/pages/ithandbook/url.asp?url=http://www.nhsia.nhs.uk/napps/step/pages/ithandbook/h235-5.htm*

[iv] Rapport de la Commission d'étude de l'accident de la Gare de Lyon, Ministère des transports et de la mer, Paris, Septembre 1988

[v] BSI Report

[vi] BSI Report

[vii] http://en.wikipedia.org/wiki/TeX

[viii] *http://www.cs.indiana.edu/docproject/programming/postscript/what-is-it.html*

[ix] *http://www.w3.org/TR/REC-html40/intro/intro.html*

[x] *http://www.w3.org/TR/html401/references.html#ref-RFC2070*

[xi] *http://www.w3.org/TR/html401/references.html#ref-ISO10646*

[xii] *http://msdn.microsoft.com/library/default.asp?url=/msdnmag/issues/03/11/xmlfiles/toc.asp*

[xiii] *http://www.microsoft.com/presspass/press/2003/nov03/11-17XMLRefSchemaEMEAPR.asp*

[xiv] *http://sunflash.sun.com/68/2/FOSS/index.shtml*

[xv] Oasis
www.oasis-open.org/

[xvi] BSI Report

[xvii] *http://koffice.kde.org/DTD/document-info-1.2.dtd*.

[xviii] *http://www.abisource.com/*

[xix] *http://www.abisource.com/awml.dtd*

[xx] *http://www.abisource.com/dev/news/2000/awn3.phtml*

[xxi] The DocBook Document Type Version 4.2b0, OASIS DocBook Technical Committee, Working Draft "Beta 0" 13 Nov 2001, *http://www.oasis-open.org/docbook/specs/docbook-4.2b0.html*

[xxii] *http://www.oasis-open.org/docbook/specs*

[xxiii] New Version of Adobe FrameMaker Extends Powerful XML Capabilities, *http://www.adobe.com/aboutadobe/pressroom/pressreleases/pdfs/200312/120103FRAMEMAKER.pdf*

[xxiv] http://www.nationmaster.com/encyclopedia/Portable-Document-Format

[xxv] The Adobe XML architecture, http://www.adobe.com/enterprise/xml.html

[xxvi] http://news.com.com/2100-1012-1025731.html

[xxvii] http://www.adobe.com/enterprise/pdfs/xml_whitepaper.pdf

[xxviii] http://support.microsoft.com/search/default.aspx?Queryc=convert+word&Query=convert+word&srch=sup&catalog=LCID%3D1033%26CDID%3DEN-US-KB%26PRODLISTSRC%3DON&Product=msall&KeywordType=ALL&Titles=false&numDa

[xxix] Aristote seminar

[xxx] StarOffice / OpenOffice.org "Q" Product Concept, Lutz Hoeger, August 2003

[xxxi] STATSKONTORET, Interoperability Test and XML Evaluation of StarOffice Writer 6.0 and Office Word 2003 Beta 2

[xxxii] https://createpdf.adobe.com/cgi-feeder.pl/help_general?BP=&LOC=en_US

[xxxiii] Aristote Seminar (Huc Presentation)

[xxxiv] http://www.microsoft.com/presspass/press/2003/nov03/11-17XMLRefSchemaEMEAPR.asp

[xxxv] Secrets of Office 2003
http://www.ecommercetimes.com/perl/story/32127.html

[xxxvi] Office 2003 XML Reference Schemas News and Reviews
http://www.microsoft.com/Office/xml/reviews.mspx

[xxxvii] Office 2003 XML Reference Schemas News and Reviews
http://www.microsoft.com/Office/xml/reviews.mspx

[xxxviii] Office 2003 XML Reference Schema Patent License
http://www.microsoft.com/mscorp/ip/format/xmlpatentlicense.asp

[xxxix] Thoughts on the Office XML Reference Schemas
http://www.adtmag.com/article.asp?id=8550

[xl] Thoughts on the Office XML Reference Schemas
http://www.adtmag.com/article.asp?id=8550

[xli] http://www.microsoft.com/office/xml/default.mspx

[xlii] Microsofts XML-skema initiativ
http://www.oio.dk/software/MS/

[xliii] http://www.microsoft.com/mscorp/ip/format/xmlpatentlicense.asp

[xliv] http://www.openoffice.org

[xlv] http://www.sun.com/service/support/software/openoffice

[xlvi] The differences between StarOffice and OpenOffice.org can be accounted for in two ways:
http://documentation.openoffice.org/user_faq/various_topics/002.html

[xlvii] StarOffice™ 7 Office Suite vs. OpenOffice.org Comparison
http://wwws.sun.com/software/star/openoffice/docs/SO_Comparison_OOo.pdf

[xlviii] Oasis
www.oasis-open.org/

[xlix] http://news.earthweb.com/dev-news/print.php/3085941

[l]

China gov't invests in open source
http://asia.cnet.com/newstech/applications/0,39001094,39156916,00.htm
Sun Notches Linux Win With Chinese Gov't
http://www.eweek.com/article2/0,4149,1385697,00.asp
Microsoft Loses to Linux in Thailand Struggle
http://www.ecommercetimes.com/perl/story/32110.html
IBM To Launch Comprehensive Linux Desktop Support Program Next Year
http://www.crn.com/sections/BreakingNews/dailyarchives.asp?ArticleID=45881
Sun and the China Standard Software Company Partner to Establish the Java Desktop System(TM) as the Foundation for China's Fast Growing IT Industry
http://biz.yahoo.com/prnews/031117/sfm143_1.html
Lindows.com Announces 30,000 Machine Deployment in Canada
Opens Canadian Office To Locally Serve Software Needs
http://biz.yahoo.com/prnews/031117/sfm143_1.html
MS scorns Israeli OpenOffice defection
http://www.theregister.co.uk/content/4/34154.html
Open source group poised to white-ant Microsoft
http://www.smh.com.au/articles/2003/11/24/1069522526315.html
Impi, Africa's First Linux Desktop
http://allafrica.com/stories/200311260385.html
Bayanihan Linux: The low-cost desktop solution for government agencies and schools
http://www.bayanihan.gov.ph/
OpenOffice.org in Uganda
http://allafrica.com/stories/200312030247.html
Vietnam and OpenOffice.org
http://www.winnetmag.com/Article/ArticleID/41069/41069.html
Janome plunges into OpenOffice
http://www.computerworld.com.au/index.php?id=281553222&fp=16&fpid=0
Linux & OpenOffice.org in Africa
http://mathaba.net/x.htm?http://mathaba.net/0_index.shtml?x=29276
OpenOffice makes government inroads
http://news.com.com/2100-7344_3-5128730.html?tag=nefd_top

[li] http://tools.openoffice.org/releases/q-concept.html

[lii] Market Overview 2003: Office Productivity Suites
Erosion Begins for Microsoft Office Dominance
http://www.gigaweb.com

[liii] Gartner, October 2003 – FT-21-3870

[liv] Gartner, October 2003 – FT-21-3870

[lv] Gartner, April 2003, LE-19-4600

[lvi] Microsofts XML-skema initiativ
http://www.oio.dk/software/MS/

[lvii] Microsoft Licenses Office 2003 XML Reference Schemas
http://xml.coverpages.org/LicenseOfficeSchemas.html

[lviii] German study by Soreon: 25% cost savings with Microsoft alternatives
http://de.internet.com/?id=2025258&section=Marketing-Statistics

[lix] Microsofts XML-skema initiativ
http://www.oio.dk/software/MS/

[lx] Office 2003: une entreprise sur cinq envisage de migrer :
http://www.01net.com/article/220532.html

La percée d'Office 2003 n'est pas pour... 2003 :
http://www.01net.com/article/220536.html

[lxi] Gartner Paper, April 2003, LE-19-4600

[lxii] OpenOffice.org XML File Format
http://xml.coverpages.org/starOfficeXML.html

[lxiii] Microsoft licensing could push users to StarOffice
http://www.computerworld.com/printthis/2002/0,4814,70710,00.html