



**UNIVERSITY OF STRATHCLYDE
GRADUATE SCHOOL OF BUSINESS**

Master of Business Administration

Assignment for Information Systems, 2005-06

Open Standards and Open Source – What about it?

Authors: Simon Axon, Student Registration No 200489732

Flexible Learning MBA – Huntly

Date: May 2006

CONTENTS

1.0	ABSTRACT	4
2.0	INTRODUCTION	4
3.0	DISCUSSION	4
3.1	WHAT ARE OPEN STANDARDS?.....	4
3.2	WHAT IS OPEN SOURCE (SOFTWARE)?.....	5
3.3	HOW ARE OPEN STANDARDS AND OPEN SOURCE LINKED?	6
3.4	USING AN OPEN SOURCE DEVELOPMENT POLICY - EXAMPLES	6
3.5	OPEN STANDARDS – ADVANTAGES AND DISADVANTAGES	6
3.6	OPEN SOURCE SOFTWARE – ADVANTAGES AND DISADVANTAGES	8
4.0	SUMMARY & CONCLUSION	13
5.0	REFERENCES	14
APPENDIX		
A1	SOME STANDARDS DEFINITIONS	16
A1.1	OPEN STANDARD	16
A1.2	DE FACTO STANDARD	16
A1.3	PROPRIETARY STANDARD	16
A2	OSS LICENSES	16
A3	LINUS TORVALDS AND LINUX	17
A4	BROWSER USAGE STATISTICS	18
A5	CODE-FORK	19

<u>Abbreviation</u>	<u>Description / Web address</u>
Adobe	www.adobe.co.uk (Software solutions company; developer of Acrobat reader)
AMD	Advanced Micro Devices (www.amd.com) (Supplier of integrated circuits for PCs)
Apache	www.apache.org (The Apache Software Foundation)
CSS	Cascading Style Sheets
Debian	www.debian.org
DNS	Domain Name System
Google	http://www.google.co.uk/ (Progressive Internet search engine), also http://scholar.google.com/
GUI	Graphical User Interface
HTML	Hypertext Markup Language
IE	(Microsoft) Internet Explorer (www.microsoft.com/ie/)
IEE	UK-based Institution of Electrical Engineers (www.theiet.org)
IEEE	US-based Institute of Electrical and Electronics Engineers (www.ieee.org)
IETF	Internet Engineering Task Force
ISO	International Standards Organisation (www.iso.org)
KDE	K Desktop Environment (www.kde.org)
Gentoo	www.gentoo.org (Linux)
GNOME	GNU Object Model Environment (www.gnome.org)
LinuxMandrake	wwwnew.mandriva.com (Linux)
Microsoft	www.microsoft.com
Mozilla	www.mozilla.org (Open source web browser)
MySQL	www.mysql.com (Open Source database)
OO	OpenOffice.org (www.openoffice.org) (Open source office productivity suite)
Opera	www.opera.com (Open source web browser)
OS	Open Source
OSD	Open Source Definition (www.opensource.org/docs/definition.php)
OSI	Open Standards Initiative (www.opensource.org)
OSS	Open Source Software
OSS Watch	www.oss-watch.ac.uk (provides Open Source guidance service)
Perl	www.perl.org (Open source stable, cross platform programming language)
pdf	Portable Document Format ('de facto' industry standard developed by Adobe)
QA	Quality Assurance
SDO	Standards Development Organisation
(E)SMTP	(Extended) Simple Mail Transfer Protocol
SSO	Standards Setting Organisation (Ketcher, 2006)
Slackware	www.slackware.com (Linux)
SourceForge	http://sourceforge.net/ (World's largest Open Source software development web site)
Sun Microsystems	www.sun.com (Computing solutions provider, both hardware & software)
SUSE	en.opensuse.org (Linux OS components only) or www.novell.com/linux (Enterprise)
TCP/IP	Transmission Control Protocol/Internet Protocol – known as a Protocol Suite
Xandros	www.xandros.com (Linux)
W3C	World Wide Web Consortium (www.W3C.org)
XHTML	Extensible Hypertext Markup Language
XLST	Extensible Markup Language Style Sheet
XML	Extensible Markup Language

Table of Acronyms

1.0 ABSTRACT

This report considers the use of open standard and open source in the development and deployment of business systems. The advantages and disadvantages of both are discussed from both a developer and deployment aspect as well as the influence of open standards on open source software development is also considered.

2.0 INTRODUCTION

This report is concerned with the use of open standards and open source in the development and deployment of business information systems. The advantages and disadvantages of Open Standards as well the OS model in a business environment are explored, from a developer, user and deployment aspect.

For the purposes of this report, open standards and open source are defined as follows:

OPEN STANDARDS

There are numerous definitions available for Open Standards, but for the purposes of this document, a standard can be a specification, a practice or a reference model. Open Standards are published standards that are unimpeded by patents or copyrights (OSS_watch, April 2006).

OPEN SOURCE

The non-profit Open Source Initiative (OSI, 2006) maintains and applies the Open Source Definition. There are a number of different aspects to the OSD concerning availability of source code in its original form, the freedom of others to use and modify the code and licensing. At the time of writing the latest definition was Version 1.9 (DiBona, OSD, 2006). Essentially the definition aims to ensure freedom to run the Open Source program for any purpose, freedom to study how the program works and adapt it to your needs, freedom to redistribute copies to help fellow citizens, freedom to improve the program, and release the improvements to the public so that the whole community benefits.

3.0 DISCUSSION

3.1 WHAT ARE OPEN STANDARDS?

A wide variety of standards are continually being developed by a number of different standards organisations, and they can broadly be split into two categories: Regulatory standards set by governments and the market driven Voluntary standards. Open Standards fall into the voluntary category and bring together all interested parties to contribute to their creation and maintenance. (For Open Standards definition, see Appendix [A1.1](#) (Coyle, March 2002).

Some key Open Standards and the organisations that make them are given in Appendix [A3](#), but perhaps the most well-known are those associated with Web standards (HTML, XHTML, CSS, XML, XLST, etc) such as those used for the author's website (www.laptech.co.uk), the World Wide Web Consortium (W3C).

Any Open Standard can be used for implementation as OS whereas 'de facto' standards – although possible, in practice are more difficult to implement; and patent protected 'proprietary' standards do not lend themselves to implementation as OS at all because patents are normally licenced by software or hardware install; OS licenses do not permit restrictions on the number of software installations.

3.2 WHAT IS OPEN SOURCE (SOFTWARE)?

There are a number of descriptions available for OS, but in general, an OSS program has its source code available for distribution to the general public together with a license allowing other programmers to alter the original software. OSS is usually developed as a public collaboration and made freely available; it does not have licencing restrictions that limit use, modification or distribution. That said there are a number of different types of licenses used, some of which are included in Appendix [A3](#). OSS might be regarded as 'free' software, but only in terms of freedom to modify, improve or distribute, not as in how much it costs.

Arguably the best-known example of OSS is Linux. Having originally developed the Linux core source code, Torvalds (Appendix [A4](#)) didn't seek payment for his work, but rather, he published it on the Internet and invited other programmers to modify and improve it. Today, there are a number of variants of Linux available, with arguably the most famous being Red Hat based. (For examples, see Abbreviation Table).

Many people browsing the Internet use OS alternatives to IE such as Mozilla and Opera, although @ 86% usage IE still dominates the web browser market by some considerable margin (Upsdell, 2006, Appendix [A5](#)). W3C produce web standards that theoretically standardise between Browsers, but it is my experience that there can be subtle differences between them when it comes to displaying web pages, hence the reason why web designers should check their web page designs using as many browsers as possible to ensure robustness of design.

At its most basic level, OSS is a low cost alternative to proprietary software. For example, OpenOffice.org (OO, April 2006) is seen by many to be a low cost alternative to Microsoft Office.

3.3 LINKING OPEN STANDARDS AND OPEN SOURCE

Open Source does not equal or imply adherence to Open Standards, however, one might suggest that open source developers are more likely to use open standards than proprietary developers on the basis that bias to open standards increases interoperability and information sharing among developers and technology providers, ultimately benefiting users.

3.4 USING AN OPEN SOURCE DEVELOPMENT POLICY - EXAMPLES

Businesses or organisations which use OS as a development platform include: Red Hat, Sun Microsystems (a sponsor of OpenOffice, which provides the foundation for its StarOffice Software Version 6.0 and subsequent versions), Apache (a non-profit organisation initially formed in 1995 to develop the Apache HTTP Server), and Mandriva (LinuxMandrake, by using a GUI for installation arguably the first truly user-friendly version of Linux).

3.5 OPEN STANDARDS – ADVANTAGES AND DISADVANTAGES

ADVANTAGES

Fair, impartial and available. Perens (No date) observes: “The key thing about open standards is that they're fair, impartial and available for everybody to implement”. Some Open Standards (such as W3C standards) are free to download, whereas some organisations might make a charge to raise funds, suggesting that such standards may not be truly open if the price prevents its use.

Compatibility. Compatibility standards “enhance or make possible technical coordination among different components of a technological system” (Antonelli 1994)

Reduces risk of customers being ‘locked-in’ to a particular vendor. Over the past decade or so, for example, Microsoft Office customers have traditionally been reluctant to change because of the dominance of Microsoft on Desktop PCs.

Competitor advantage. Finally, even with full disclosure of all formal interfaces, there are still opportunities for sponsors to gain advantage through their knowledge of tacit information gleaned through the creation of a standard. For example, Sun Microsystems originally developed StarOffice before disseminating ‘de facto’ standards that were made available to customers, complementers and competitors. However, such as with a fully proprietary standard such as Microsoft Office, Sun’s control and knowledge of the technology gave it the quickest access to specifications and thus a time advantage in creating implementations. One might also argue that it benefited from their developer’s familiarity of

the standard through 'learning by doing'. Sun's StarOffice Suite Version 8 reflects additional features and quirks to OO, on which it is based.

DISADVANTAGES

SDOs are too slow and unresponsive. West (2004) comments that formal SDOs are widely seen as being too slow and unresponsive to provide anticipatory standards in industries with rapidly changing technologies, although he acknowledges that this criticism is rejected by many who participate in the formal standards effort. That Microsoft and Adobe have created their own standards (for example, 'doc', 'xls', 'ppt' in the case of Microsoft, and 'pdf' (Adobe)) has meant that they can develop their own software to suit the development of their own 'de facto' standards without involving external organisations. Indeed, Adobe still owns the 'pdf' specification, which originated in the early 1990s and was considered for submission to a standards body. Becker (2004) records that Tim Bray (co-inventor of XML) observes that "'pdf' is an example of a proprietary standard that has achieved so much inertia in the marketplace that it's hard to see where standardizing it would benefit anyone much".

Open Standards are exclusive club. West (2004) also observes that Open Standards organisations might be seen as an exclusive club, and that a key measure of the degree of openness of any group of companies creating the standard (whether a SDO, consortium or formal alliance) is the degree of access available to non-members. If non-members have the same use rights as members, then the standard itself could be regarded as available to all, whereas if non-members have little or no rights, then the standard is only of benefit to its members or creators. One might also wish to investigate the membership policy of the standards group, and its openness to non-members. A closed group might include two powerful companies such as Microsoft and Intel, both companies being dominant in their respective areas of industry (noting AMD as Intel's main competitor), in other cases there may be pre-defined membership requirements, such as with the IEE, or there may be cases where there is true openness and the process is open to any interested party such as the IETF. Krechmer (2005) observes that there are three types of Open Standard:

- Work-in-progress documents are only available to committee members (standards creators). Standards are for sale. (Current state of most formal SSOs.)
- Work-in-progress documents are only available to committee members (standards creators). Standards are available for little or no cost. (Current state of many consortia.)
- Work-in-progress documents and standards are available for reasonable or no cost. (Current state of IETF.)

Open Standards reduce vendor power over buyers. This might be perceived as an advantage to buyers or consumers, but a disadvantage to businesses. For many businesses, an open standard might be one that reduces vendor power over buyers, usually due to competition between implementations of a standard and the availability of alternative but compatible products.

3.6 OPEN SOURCE SOFTWARE – ADVANTAGES AND DISADVANTAGES

ADVANTAGES

OSS developers are highly motivated. There must be a reason for developers to participate in and contribute to open source projects. Gustafson & Koff (Sept 2004) report that in a survey of 684 developers (Figure 3.6 below); the primary motivation was that the work was intellectually stimulating, and the work improves skill and work functionality (Figure 4.6). All of these are high in Maslows Hierarchy of Needs (Mullins, 2005). The heart and soul of OSS projects are their community; Maslow claims that the hierarchy is relatively universal among different cultures, and this might go some way to explaining how OSS projects can span countries and continents to great effect and be so successful. Therefore, the first advantage is that the developers are highly motivated. This is part of ‘nerd culture’, which thrives on the satisfaction of authorship and the respect of one’s peers rather than wealth and material items. Weber (2004) expands on this by suggesting that there are several facets to this motivation, including art and beauty, treating the job as a vocation, consider manufacturers of proprietary software (especially Microsoft) as the joint enemy, the work can also be ego boosting (as well as damaging if, say a developers code is rejected from the project), and can enhance reputation, especially if a developers code is well received in an OSS community. Raymond (2001) says as much in “*The Cathedral and the Bazaar*”, his musings on Linux and open source.

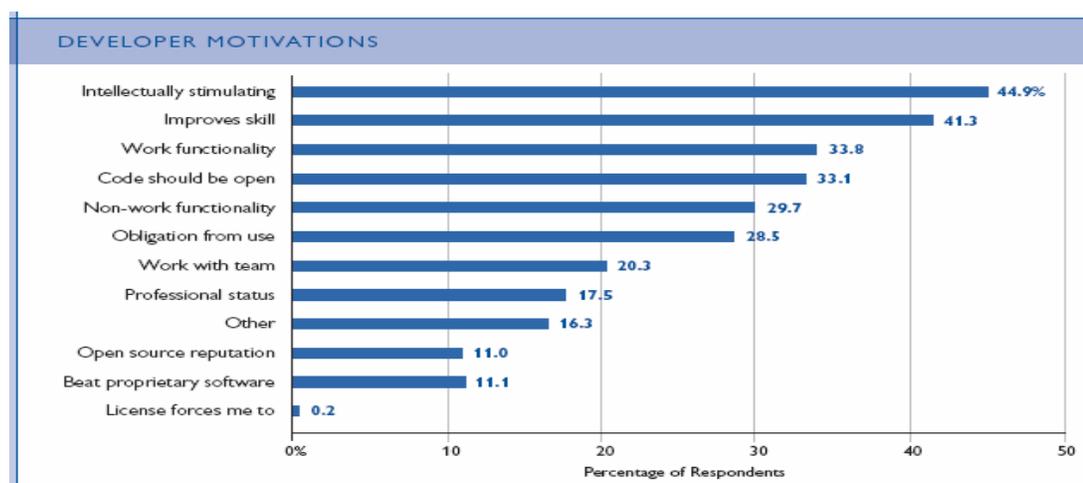


Figure 3.6: OSS Developer Motivations

Source:

The Boston Consulting Group, in cooperation with the Open Source Developer Network, surveyed 684 developers participating in software projects on SourceForge.net (2002).

OSS code availability. The availability of source code and that any developer has a right to modify the code means that the code is available for any developer to improve. Source code availability makes it much easier to isolate bugs in the software and for a developer to fix them.

OSS can be modified and redistributed, offering customisation. Combined with redistribution rights, that OSS can be modified and used in anyway can help ensure a larger user base, and this in turn helps build a market for support and customisation of the software. Potentially this can help attract more developers onto a project usually helping improve the quality and functionality of the product. If the quality and functionality of the product is well received, it can become well known and users might be keener to give it a try. The right to freely modify OSS can lead to significant and rapid expansion in the development of new technologies, increasing knowledge at minimal cost. Customisation means being able to improve and audit the programs. Project stakeholders will have to decide how much they can afford to invest in such efforts, but OS makes the decision theirs rather than a vendor. With OS allowing room for modification and distribution, business (Government, etc) is able to customise solutions according to its specific needs.

OSS can offer more flexibility and freedom from (operating) platform lock-in. For example, OO will run on older Windows™ platforms as well as Linux based Operating Systems. This is because the OSS community elected to give their software as much flexibility as possible. Microsoft Office 2003 or XP™ will only run on Windows 2000 or later, although there is a version of Microsoft Office™ that will run on Applemac. There is no variation of MS Office than will run directly on Linux based Operating Systems, although there is emulation software available, which will allow users to run some Windows based software on Linux based operating systems. There is also a version of MS Office that will run on Applemac. Open source software (OO, for example) is typically available on dozens of platforms, allowing a business to choose the most economical combination of hardware and software for its needs. By providing ready access to source code, open source ensures freedom, thereby preventing a business being tied to a single company or platform and potentially ensuring access to critical data in all situations.

OSS can offer quality due to code availability, openness and transparency. The resources an open source software project can be huge, and dwarf what a commercial effort - compensating its developers with money, can offer. Thus, open source software can be exceptionally well-designed, robust, bug-

free, and amenable to user requests for enhancement. OS developers are not constrained by corporate product development processes or ISO 9001-style software development and QA processes. In October 2001, Peeling & Satchell (2001) reported that although some Open Source software is unreliable, many of the most popular OSS products have a rate of evolution, robustness and responsiveness to bug reports. I suspect that the OSS development has moved on a lot from then, where products have become a lot more stable and robust.

No exclusive rights. When no one holds exclusive rights to code, no one has the power to restrict how the software is used in a singular way. One of the frustrations with proprietary software vendors such as Microsoft is that because they are so dominant, when they release new products they can choose to restrict its use on older operating platforms. As a result, customers can only stick with the older software version or switch products. OSS potentially gives the opportunity for the user to look to other Vendors to provide upgrades of the same product (e.g. Linux, of which there are several variations).

OSS offers protection against non-availability. Users or customers may rely solely on a single software manufacturer for a particular software product. If the manufacturer collapses, the program is most likely to remain dormant and not updated due to licensing restrictions. Theoretically, OSS protects against this by being available for further development without legal implications, assuming that funding or motivation can be found.

Commercial priorities less invasive. Since there is no commercial entity pushing for exact delivery dates or features that must be included, the implication might be that any software can be made more complete when released as it might not need as many service packs (cue Microsoft) and updates, thereby reducing software servicing costs. There are limits, of course. OSS couldn't be delayed indefinitely as it would probably fail to meet the expectations of the users.

Transparent scope for improvement. The scope for improvement of OSS is perhaps a lot more transparent than for proprietary software. Individuals, communities or committees might decide the improvements and direction that the development might take, or peer pressure might decide overall development of progress. However, where conflict occurs, the product might be split where different OSS communities take the development of software in different direction, for example by forking (Appendix [A5](#)).

Ease of Deployment. Open source software tends to focus on the most essential and used capabilities, rather than having a large quantity of rarely used features. This makes deployment easier than proprietary software. Further, users are able to deploy the software without having to sign licenses, or

make financial cases to management although there are other issues, such as cost of training, etc. as identified below. More and more, mature open source software suitable for mass installation comes with easy-to-use installation software, graphical management tools, and on-line help. OO is a good example of this.

Flexibility. Customers can have the flexibility to use internal skills or other service providers for maintenance and support instead of depending on one company that owns the software.

Lower cost of operation. Open source products typically not only have lower license costs, but can also dramatically reduce the running costs that make up the total cost of ownership. For large organizations, this can result in significant operational cost savings, although as mentioned earlier, training and development costs would need to be accounted for.

Security. Because open source software is out in the open, the open source community will argue that it is typically more secure and suffers fewer vulnerability attacks than proprietary software, although this is disputed by Perens (No date). Further, when a problem is uncovered, it can be addressed quickly.

Increasing acceptability. Use and acceptability of OSS is becoming increasingly common on servers, networks and appliances with the advent of such OSS software such as Linux, Apache, MySQL, and Perl.

Availability. OSS can be available from a number of different software vendors.

Focus on services. Some might argue that with OSS there tends to be a focus on services and what the customer wants, rather than say on license fees or bloated software.

DISADVANTAGES

Potential security loopholes. In his article "*The Trojan Horse*", Perens (1998) asks if users really know where their software comes from, and can they really be sure that the software is virus or bug free, and does not contain a back-door trojan (given that code could be several thousand lines long). In his article, Perens suggested ways in which the source code can be tightened up in order to avoid security loop holes arising.

Training and development costs. Low acquisition cost is not the best reason to decide on OSS source software. The software itself is usually free, but training maintenance and support might cost as much

or even more as they would with proprietary software on the basis of unfamiliarity with the product for example when making a transition from using say Microsoft Office to OO.

Resistance to change. Zuliani & Succi (2004) report in the experimental deployment of OO as a viable alternative for Microsoft Office, within several municipalities of the Province of Bolzano-Bozen, Italy. However, they suggested that the biggest problem deploying OO was resistance of personnel to change, with training and development being considered as adding to an employees work burden.

OSS project may not progress to completion. Theoretically an OSS project might not progress to completion due to say lack of motivation, lack of resource, lack of funding or lack of direction. Theoretically, this could happen with the development of proprietary software, but arguably this is more likely to happen with the development of OSS. Barahona (2004) suggests that OS can reach a point where a project is at a self sustaining level, where the user and development base is such that it can proceed by itself without other external incentives, but clearly the project would need some form of direction if it to be developed to a level where it can be deployed.

Intellectual property may be an issue. As more and more patent rights are given for specific software intellectual property may become a problem as more and more countries accept software and algorithm patents. Over time, specific problems may be offset or simplified by the availability of source code which would enable detection of patent infringements by patent holders. The absence of a single company that holds all the rights on the software developed by an OSS community can also make it difficult for companies to defend from patent litigation, like cross-licensing or payment of royalties.

Lack of progress transparency. It is the author's experience that unless one is heavily involved in an OS project it is difficult to gauge how well progress is going. That said while it doesn't really give a quantitative measure of progress, Linux projects might typically have two versions of a kernel available. One would be a 'stable version' which has been subject to more rigorous testing by the development community, which a user could download and install, but also as a more experimental and more recent version in the process of development, not so rigorously checked, which a user can also download and install, but with much more risk of it being affected by a bug. In many instances, however, lack of transparency might simply be due to the constant evolving nature of such projects.

Proprietary solutions are superior. Many contend that proprietary solutions are technically superior to OS because they have dedicated developers or because they are most expensive; others argue that those developing OS solutions do so out of personal interest and are therefore extremely motivated to produce the best, most stable and secure software possible.

Misconception of OSS deployment possibilities. The belief that just because the software is free does not mean that one cannot charge for the installation.

Lack of OSS familiarity. End users 'in the office' are currently still more familiar with proprietary applications such as Microsoft Office as opposed to OS applications such as OO or Linux. That said, within processes in the back end and server environments, OSS has already made a huge impact and depending on the application often competes or co-operates very successfully with proprietary applications. For example, according to Netcraft's Web server survey apache runs nearly 70 percent of over 50 million Web sites surveyed (Gustafson & Koff, Sept 2004).

Permanent possibility of code-fork. Open source (software development) businesses must face the permanent possibility of a code-fork (Appendix [A5](#)). They lessen this risk by engaging with the open source community directly and honestly, working to build that community and strengthen it.

Lack of technical support for OS. The reality is that while most people are unaware of it; everyone who has access to the internet has probably made use of OSS via servers using Linux, or Apache, for example. The search engine Google uses servers that are run using Linux. The OS community has a tradition of being very helpful via web forums; the biggest hassle might be in describing your problem or the help you need, couched in a language that is easy to interpret and understand. While technical support via telephone may be difficult if not nigh on impossible for OS projects, it is the author's experience that telephone support offered by proprietary vendors can also be very frustrating while one waits at the end of the phone for a reply.

4.0 SUMMARY & CONCLUSION

Open Standards are published standards that are unimpeded by patents or copyrights. OSS has its source code available for distribution to the general public together with a license allowing other programmers to alter the original software. OS does not equal or imply adherence to open standards, but open standards may be implemented by software developed under any development and licensing model - non-OSS and OSS alike.

The main advantage of Open Standards to business is their functionality, while the disadvantages are that they might not always be as open as they are meant to be, potentially stifling some of their functionality. The main advantages of OSS are potentially low cost of operation, the motivation of its contributors as well as flexibility, customisation and freedom from Vendor reliance. The main disadvantages are the perceived lack of technical support, as well as hidden costs for training and

development. Intellectual property of OS could be an issue, depending on the complexity of the software and how it was derived.

Word Count from “Introduction” to the word “derived”: 4000.

5.0 REFERENCES

Antonelli, Cristiano, (Dec 1994) “*Localized technological change and the evolution of standards as economic institutions*,” Information Economics & Policy 6, 3-4 p195-p216.

Ball, B & Duff, H. (2003) “*Red Hat Linux 9*”, SAMS Publishing

Barahona, JMG (et al), (2000) “*Free Software / Open Source: Information Society Opportunities for Europe?*”, Information Society Directorate General of the European Commission. Document updated from time to time on www.eu.conecta.it

Becker, D. (June 2004) “*When standards don’t apply*”, CNET News.com

Upsdell, C. (6 May 2006), “*Browser News*”, 398th Edition

Coyle, K. (March 2002), “*Open Source, Open Standards*”, Information Technology and Libraries, March 2002 21,1

DiBona, C. et al (2006), “*Open Sources 2.0*”, O’Reilly

Gates, B. (1998), “*Compete, Don’t Delete*”, The Economist, June 13, 1998.

Gustafson, P. & Koff, W. (Sept 2004) “*Open Source: Open For Business*”, CSC Leading Edge Forum

IEE (April 2006) Institution of Electrical Engineers www.theiee.org

IEEE (April 2006) Institute of Electrical and Electronics Engineers, Inc www.ieee.org

ISO (April 2006) The International Organization for Standardization

<http://www.iso.org/iso/en/ISOOnline.frontpage>

ITEF (April 2006) “*The Internet Engineering Task Force*”, www.ietf.org

Humphreys, P. (June 2002), “*Linus Torvalds – Master Programmer*” Virtual Finland

Krechmer, K. (Feb 2005), “*Open Standards Requirements*”, International Center for Standards Research, University of Colorado

Mandriva (April 2006), www.mandriva.com

Mullins, L. (2005), “*Management and Organisational Behaviour*”, FT Prentice Hall

Nam, T.J. (Sept 2005) “*Business Case for Open Source*”, Novell

OASIS (April 2006), www.oasis-open.org

OSI (April 2006) “*The Open Source Initiative*” www.opensource.org

OO (April 2006) “*OpenOffice.org*” www.openoffice.org

OSS_Watch (April 2006), www.oss-watch.ac.uk.

Perens, B. (1998), “*The Trojan Horse*”, Linux Weekly News

Perens, B. (1999), “*The Open Source Definition*”, from Open Sources 1.0: Voices from the Open Source Revolution”, O’Reilly

Perens, B. (No Date), “*Open Standards Principles and Practice*”
<http://perens.com/OpenStandards/Definition.html>

POST (June 2005), “*Open Source Software*”, The Parliamentary Office of Science and Technology,
Postnote Number 242

Raymond, E. (2001), “*The Cathedral & the Bazaar*”, O’Reilly

Red Hat (2006), www.europe.redhat.com

Rosen, L. (2005), “*Open Source Licensing – Software Freedom and Intellectual Property Law*”,
Prentice Hall

Sliman, Erik (May 2002), “*Business Case for Open Standards*”, www.openstandards.net

Star Office Suite (April 2006), <http://www.sun.com/software/star/staroffice/index.jsp>

The Apache Software Foundation (April 2006), www.apache.org

Torvalds, Linus (April 2006) www.cs.helsinki.fi/u/torvalds/

W3C, April 2006 World Wide Web Consortium www.w3.org

Weber, S. (2004), “*The Success of Open Source*”, Harvard University Press

West, J. (May 2004), “*What are Open Standards: Implications for Adoption, Competition and Policy*”,
Paper, presented at Standards and Public Policy Conference, 13 May 2004.

Zuliani, P. & Succi, G. (2004), “*Migrating Public Administrations to Open Source Software*”, Free
University of Bolzano-Bozen

APPENDIX

A1 SOME STANDARDS DEFINITIONS

A1.1 OPEN STANDARD

According to Coyle (2002), Open Standards are:

- Standards that anyone can use to develop software or functions
- Standards in which anyone can participate in their development and modification, and
- Standards that anyone can obtain without a significant price barrier

Essentially, Open Standards are published standards not affected by patents or copyrights and are affordable to all.

A1.2 DE FACTO STANDARD

A 'de facto' standard is a format, language, or protocol that has become a standard not because it has been approved by a standards organization but because it is widely used and recognized by the industry as being standard. Some examples of 'de facto' standards include:

- Adobe portable document format (pdf)
- Microsoft doc, xls, ppt extensions for Microsoft Office documents
- Xmodem Communications Protocol
- Hewlett-Packard Printer Control Language (PCL) for laser printers
- PostScript page description language for laser printers

Ref: http://www.webopedia.com/TERM/D/de_facto_standard.htm

A1.3 PROPRIETARY STANDARD

A Proprietary Standard is a specification for hardware and software that are developed and controlled by one company. Proprietary standards are technically de facto standards such as Microsoft's Windows and Intel's x86 chip family (or AMDs chip families).

<http://www.techweb.com/encyclopedia/defineterm.jhtml?term=proprietarystandards>

A2 OPEN STANDARDS AND ASSOCIATED ORGANISATIONS

Some other key Open Standards and the organisations that make them are given below:

- Core Internet standards (TCP/IP, (E)SMTP, DNS, etc.) — the Internet Society's Internet Engineering Task Force (IETF)
- Underlying electronic communications standards (Ethernet, Firewire, WiFi, etc.) — the US-based Institute of Electrical and Electronics Engineers (IEEE) and the UK-based Institution of Electrical Engineers (IEE)

- General international standards (programming languages, mark-up, Unicode, natural language character sets, etc) — the International Standards Organisation (ISO) and the 148 national standards bodies affiliated to it

A3 OSS LICENSES

Some of the more common OSS licenses (Rosen, 2005 (uno)) that the author is aware of are given below:

- Berkeley Software Distribution (BSD) licence: this permits the licensee to ‘close’ a version (by withholding most recent modifications to the source-code) and sell it as a proprietary product;
- GNU General Public Licence (GNU GPL or GPL): under this licence, licensee may modify, copy and redistribute any derivative version, under the same GPL licence. The licensee can either charge a fee for this service or work free of charge;
- GNU Lesser General Public License (GNU LGPL) as used by OO (www.openoffice.org, April 2006). This license is called the "Lesser" General Public License because it does less to protect the user's freedom than the ordinary General Public License. OO is a multiplatform and multilingual office suite and an open-source project. Compatible with all other major office suites, including versions of Microsoft Office TM the product is free to download, use, and distribute, and has been installed and used successfully by the Author.
- MPL (Mozilla Public License). This is the licenses made by Netscape to distribute the code of Mozilla, a newer version of its network navigator. In many respects this license is similar to the GPL, but is perhaps more ‘enterprise oriented’.

A4 LINUS TORVALDS AND LINUX

According to Patrick Humphreys (June 2002), the story of Linux begins in 1991, when Linus Torvalds, a 21-year-old student at Helsinki University, decided to write his own computer operating system. When Linus Torvalds first developed Linux back in August of 1991, the operating system basically consisted of his kernel and some GNU tools. With the help of others Linus added more and more tools and applications.

In some respects, some argue that Torvalds’ Linux is better than the world’s principal operating system, Windows. It is more compact, runs faster and it is more stable, so it is preferred for use on the Internet, powering web servers that can be left unattended, without operatives to "turn off and then start again", as Windows so often requires.

Linux is the outcome of a completely different philosophy, making its author a cult figure. Torvalds did not claim his right to receive payment from people who used his work. Instead, he published it on the Internet and invited other programmers to improve on it and to send their results back to him by email.

Linux remains a free program. Anyone can use it without charge, on condition that any improvements they make are also freely available to other users. The computer nerds of the world took up Torvalds' challenge. Of Linux today, Linus himself only wrote about 2%, although he remains the ultimate authority on what new code and innovations are incorporated into it.

This is quite a contrast with Windows, the operation of which is a proprietary Microsoft secret. Because the original quantities and instructions that make up Linux have been published, any programmer can see what it is doing, how it does it and, possibly, how it could do it better. Torvalds did not invent the concept of open programming but Linux is what made it truly visible. Indeed, it probably could not have succeeded before the Internet had linked the disparate world of computing experts.

In making Linux an open language, Torvalds gave up the opportunity of growing rich from his work. This too is part of nerd culture, which thrives on the satisfaction of authorship and the respect of one's peers rather than wealth and material items.

A5 BROWSER USAGE STATISTICS

Note that the following statistics are subject to browser detection capability and there maybe a number of browsers which are undetected; without access to log files there might be no way of knowing what they are. Text browsers may be undercounted, and a meaningful and valid sample needs to be available from the website monitoring the browser usage. Other problems might be related to the configuration of the Internet Browser, for example Opera can be configured to be identified as Microsoft Explorer.

- Gecko-Based Browsers (Camino, Firefox, Mozilla, Netscape 6-8, SeaMonkey, etc.): • The reported percentage of users varies a lot, largely because different sites attract different types of users. • I suggest that ~11% typically use Gecko browsers, but many more use Gecko browsers on sites that attract people who are more aware of the alternatives.
- KHTML-Based Browsers (Konqueror, OmniWeb 4.5+, Safari, etc.): • I suggest that ~2.5% typically use KHTML browsers — the great majority being Safari users — with the number likely to rise slowly as more OS X users switch to Safari, especially since Microsoft has abandoned IE for Mac operating systems: however, the fact that only ~4% use OS X puts a cap on how many will eventually use Safari.

- Microsoft Internet Explorer: Roughly 83% use IE-based browsers, down from a high of ~94% as users switched to other browser families — mainly Gecko and KHTML. • Most use IE6, with its number increasing very slowly as users switch from IE5. • Many use IE5, with the number shrinking as users upgrade or switch; IE5 will remain a factor for several years. • Few use IE4, <.1%, with the number shrinking steadily. For most sites the number of users is negligible.
- Netscape 4.x: • Stats vary, but <¼% likely use a Netscape 4 browser; the number is shrinking slowly and may not become negligible till the end of 2005. • Some sources sometimes report anomalously high numbers of users. This may be due to search engines that identify themselves as Netscape 4.
- Opera: • The reported percentage of users varies a lot, and is likely under-reported because many stats sources wrongly identify Opera as Internet Explorer, Mozilla, or Netscape; one survey in Mar 2006 revealed that 76% of Opera users had configured it to identify it as Internet Explorer. • I suggest that ~1% typically use Opera browsers, with the number changing little. • The great majority use Opera 7 and 8: about 1% use older versions.

Source: Browser News No. 398, http://www.upsdell.com/BrowserNews/stat_trends.htm, 6 May 2006

A6 CODE-FORK

In software engineering, a project fork or branch happens when a developer (or a group of them) takes a copy of source code from one software package and starts to independently develop a new package. The term is also used more loosely to represent a similar branching of any work (for example, there are several forks of the English-language Wikipedia) particularly with free or open source software.

Forks with free software result from a schism over different goals or personality clashes. In a fork, both parties assume closely identical copyrights but typically only the larger group or that containing the original architect will retain the full original name and its associated user community. Thus there is a reputation penalty associated with forking.

Source: *Wikipedia*, the free encyclopedia. [http://en.wikipedia.org/wiki/Fork_\(software_development\)](http://en.wikipedia.org/wiki/Fork_(software_development))