

MNCC OSSIG Awareness Sub-Group Paper

The Case for Open Source: OSS vs Proprietary Software

By: Tan Tze Meng

Version: 1.2

Date: 14th August 2003

Revision History

Original Author: Tan Tze Meng, Awareness Sub-Group Leader, MNCC Open Source Special Interest Group
Creation Date: 15th August 2002

Version 1.0	Date: 24 th August 2002	Modified by: Tan Tze Meng, Michael Choo
Version 1.1	Date: 27 th June 2003	Modified by: Tan Tze Meng
Version 1.2	Date: 14 th August 2003	Modified by: Tan Tze Meng

Table of Contents

Objectives.....	4
Primer.....	4
Introducing Open Source.....	6
The Cost of Proprietary Software Solutions.....	8
Open Source Adoption in the Industry and Governments.....	11
Vendor Support for Open Source.....	13
The Open Source Advantage.....	14
Advantages for End Users.....	14
Advantages for Developers.....	16
The Open Source Dilemma.....	19
Common Misconceptions About Open Source.....	19
Other Reasons Slowing the Adoption of Open Source Solutions.....	20
Issues Specific to Linux as a Special Case.....	22
Closing.....	23
Appendix 1: Sample of Open Source Solutions.....	24
Internet Server Applications.....	24
Database.....	24
Application Servers.....	24
Security Software.....	24
Network Management Tools.....	25
File and Print Server Applications.....	25
Operating Systems.....	25
Client Software.....	25
Office Applications.....	26
High Performance Cluster Computing Software.....	26
Desktops.....	26
CAD and Graphics.....	26
Software Development tools.....	26

A Case for Open Source OSS vs Proprietary Software

Objectives

The objective of this paper is to introduce the concept of Open Source to IT Managers of Businesses and to encourage the adoption of Open Source solutions through the provision of unbiased information on the subject.

Primer

To understand the concept of Open Source, one must first understand what Source is.

Source is the shortened form of the term Source Code. Source Code is a textual, human readable form of a computer program and is typically written by a human programmer.

A computer program is simply a step by step list of explicit instructions written in a programming language by the programmer to instruct the computer in performing a task. Unlike human languages, programming languages have a strict syntax to minimise ambiguities. Like human languages, there are many programming languages.

The following is a simple example of source code instructing a computer to add two numbers and display the result:

```
a:=100;  
b:=50;  
c:=a+b;  
write(c);
```

Before a computer can act upon these instructions, the program must be converted from the human readable programming language form into a computer instruction form which is called executable code, machine code or

binary code. Executable codes are sequences of numbers which are meaningless to humans. This process of conversion is termed “compiling”.

Purchased software is almost always executable code and the cost of the purchase entitles the purchaser the “right to use” of the software. The purchaser does not “own” the software. Users of purchased software almost never get access to Source code. Most proprietary software producers consider their Source Code to be their “trade secret” and kept private and proprietary. This is termed “Closed Source” software.

Software products where the Source Code is freely available is termed “Open Source”.

Introducing Open Source

To most people Open Source means free software. However, there is more to Open Source than just free software. Simply put, Open Source means software with freely available source code (*software in human readable text form, before compilation into machine executable binary form*). The source code is freely “open” for many purposes. Free software without available source code is not Open Source. Open Source is at its heart a form of software licensing with the following common key principles:

There is no license fee charged for the software

The source code is freely available (*however this does not necessarily mean zero cost*)

There are no restrictions on the use of the software, even for commercial purposes

Derivatives of the software or “greater works” of which the software is a component are allowed and encouraged with credit given to the original author(s)

Although possible, there are no known cases of litigation following violations of such licenses.

Open Source is also a collaborative software development model where anyone can participate. The people contributing towards any Open Source project are collectively referred to as the Open Source Community. This Community has no boundaries and contributors come from all over the world and many who work together on projects may have never met physically.

A full definition of Open Source is on the Open Source Initiative's web site <http://www.opensource.org>.

There are over thirty types of Open Source license (*approved by the Open Source Initiative*) but most are variants of the Free Software Foundation's GPL (*GNU Public License*) or the BSD (*Berkeley Standard Distribution*) license. The two differ in many details but one key point is that derivatives of GPL software must also be GPL whereas BSD type licenses have no restrictions on the use of the source code including the commercialisation of the product or derivatives.

So who writes all this software and gives it away for free ? Open Source software comes from many origins, Non-profit organisations such as the Free Software Foundation, FreeBSD Foundation and Apache Foundation; research and educational institutions, individuals and loosely organised groups of individuals. All these have one common principle: Software is knowledge and knowledge should be freely available to anyone who needs it.

Open Source software also comes from commercial entities such as Apple Computer, Sun Microsystems and Red Hat and forms part of their business strategy. These vendors have realised that the Open Source development model has advantages that outweigh any perceived advantage of keeping software proprietary. Many software businesses also provide Open Source software as scaled down versions of their top-of-the-range products and many have business models which generate revenues from services such as consultancy, training and customisation rather than direct software licensing.

There is a wide range of Open Source software which covers everything from operating systems to specialised applications such as the following sample:

Operating Systems: various BSD UNIX variants such as FreeBSD, Darwin (*the core of Apple's Mac OS X*), NetBSD and OpenBSD, and various Linux distributions such as Red Hat, SuSE and Debian;

Server software: BIND DNS server, Apache web server, exim MTA (*message transfer agent*) server, Darwin streaming server, Samba SMB/CIFS file/printer sharing server; OpenLDAP directory server; MySQL and PostgreSQL database servers; jBoss and Tomcat J2EE application servers;

Office applications such as OpenOffice office application suite; Koffice office application suite; AbiWord word processor; Gnumeric spreadsheet

Specialised tools such as BLAST molecular sequence database search tools and cluster computing software such as MPICH (cluster message passing solution) for Beowulf High Performance Computing clustering systems.

These are just a small sample of the tens of thousands of Open Source software available.

The Cost of Proprietary Software Solutions

Typical business solutions have a minimum set of components which are required to make things work well. These typically represent the following costs to the overall solution:

System Hardware (<i>server hardware and clients</i>)	30%
Network Hardware (<i>switches, routers, structured cabling</i>)	10%
System Software (<i>operating systems, clustering, backup</i>)	5%
Support Software (<i>DNS, web, LDAP, mail etc</i>)	10%
Application Software (<i>databases, application servers</i>)	25%
Security Hardware (<i>firewall & log server, IDS sensors</i>)	5%
Security Software (<i>firewall, scanners, IDSes</i>)	5%
Implementation Services (<i>deployment, configuration</i>)	10%

Note:

The breakdown of project costs are derived from averages of projects undertaken by the author over the course of his career. There will be variations depending upon project specifics and these figures are just to provide a general feel.

Support infrastructure costs such as external network connectivity, UPS, power supply, generators, air conditioning, clean-room etc are excluded.

Of these costs, software components make up 40% of the overall solution cost which means software licensing is the single biggest component. In many cases software costs grow with business growth as many vendors charge licensing fees based on criteria such as number of users, data records, email accounts, server class, number of processors and even processor speed. Many also charge annual maintenance fees, typically 15-25% of license fees to keep software up to date and receive support and bug fixes. All these additional costs prolong or even extend the ROI period for any solution as the business grows. Apart from the financial costs there are intangible costs which come from vendor lock-in due to proprietary software and planned obsolescence which forces constant upgrades by terminating support for old versions.

The solution to controlling software costs is the deployment of Open Source software in as many components as possible, if not all. Except for complex vertical applications, Open Source software is probably available for most requirements.

Consider the following simple example:

A typical office environment with a mixture of clients with a common file and print server.

Desktop Systems (100% Open Source)

Operating System: FreeBSD, Darwin, OpenBSD, NetBSD or Linux

GUI – X windows based: KDE2, Gnome, WindowMaker

Office Application: OpenOffice

Web Browser/Email client: Mozilla

Desktop Systems (Partial Open Source)

Operating System: MacOS X, Windows

Office Application: OpenOffice

Web Browser/Email client: Mozilla

File/Print, email Server

Operating System: FreeBSD, Darwin, OpenBSD, NetBSD or Linux

File / Print Server software: Samba SMB/CIFS/CUPS file/print server,

Mail server software : sendmail, popper

The greatest savings here would be the replacement of typically a windows 2000 based file and print server with Microsoft exchange mail software (*which is licensed on a per user basis*) and the replacement of Microsoft Office with OpenOffice on all the desktops. These cost savings are of greatest significance to large corporations, government organisations and educational establishments where large numbers of licenses are required.

Cost savings become more significant when larger servers are examined such as ISP servers:

The most basic of ISP services is email. Large ISPs will need servers capable of handling 1 million or more email users. Such large scale email servers usually consist of a user database in an LDAP server, a message store, mail transfer agents as well as POP, IMAP and Webmail. Proprietary email software licensing can cost between US\$1 (*heavily discounted for large scale customers*) to US\$5 per mail user and requires a maintenance charge of 25% of the license fees after the first year. This means a 1 million user mail server will cost US\$1 Million for the first year and US\$250,000 for subsequent years.

Open Source mail software such as sendmail, exim, qmail, perdition mail retrieval proxy and various pop3 and imap4 servers configured in a horizontally scalable architecture (*many small systems instead of few large servers*) are quite capable of being used in such a large scale email system with immense savings on license fees.

In cases where Open Source software does not meet all requirements, it is still possible to make some savings by using Open Source components with proprietary components. For example, many content management software will work well with Apache web servers. The key point here is that Open Source solutions can be used in conjunction with proprietary software and since Open Source software tends to be platform independent (*based on portability standards such as POSIX*) many will run on proprietary operating systems such as Sun's Solaris, IBM's AIX, HP's HP-UX, Compaq's Tru64, SGI's IRIX and even Windows (*using RedHat's CygWin, a UNIX like POSIX compliant API for Windows*).

In addition, many of these proprietary operating systems are bundled with Open Source software, Apache web server and OpenSSH secure shell are the most common.

Open Source Adoption in the Industry and Governments

Many businesses use Open Source software with great success. These are not small scale implementations by small business but some of the largest net-economy and media businesses as well as traditional old world economy companies today such as:

Amazon.com	NTT Docomo	Cisco	Akamai	NASA JPL
RedHat	Yahoo	Apple Computer	Hotmail	Deutsche Telekom
CDROM.com	Pixar	Etrade	7-Eleven	Sony
DreamWorks	Boeing	IBM		

Many governments are also in various stages of leveraging Open Source to reduce cost as well as to improve the “openness” of solutions. Examples which have been “in the news” recently are:

China	Korea	Germany	United Kingdom	Malaysia
Singapore	United States	European Union	Peru	Taiwan

And the list of corporations and governments grows steadily. More details of some of these implementations are below.

Amazon.com saved US\$68 Million by replacing proprietary hardware, OS and web server software with low cost Intel based hardware, Linux and Apache web server software.

Source: NetworkMagazine Jan 2002

Boeing Inc, R&D Division implemented a new Aircraft Design Application by replacing a proprietary solution with Linux Beowulf HPC Cluster based solution and gained a 400% productivity improvement at 1/3 the cost of the previous solution.

Source: InfoWorld

The Singapore Civil Service is considering replacing MS Office with an Open Source solution. Microsoft's latest Office XP Standard retails at S\$889 and even at a 70% discount, the government would save S\$27 million if it were to switch from MS Office to StarOffice.

Source: Business Times Singapore

China, Xinhua news agency reported high prices and security flaws as the main issues with Microsoft's products. Beijing chose a homegrown Linux system, Red Flag, and office applications software costing US\$1.96 billion over five years. The equivalent Microsoft OS and Office suite would have cost at least US\$21.7 billion over the next five years.

Source: Business Times Singapore

South Korea

The Central Procurement Office announced a plan to migrate its office applications and operating systems away from Microsoft. The deal involves Hancm, a local Linux distributor supplying the Korean government 120,000 copies of the Linux operating system, and a Korean-language MS Office-compatible suite, HancmOffice. This would save the Korean government up to 80% of what an equivalent purchase from Microsoft would have cost.

Source: Business Times Singapore

Taiwan

The Taiwanese government plans to start an Open Source project as early as next year that could save it as much as \$295 million in royalty payments to Microsoft.

Source: Taiwan Central News Agency

United Kingdom

The UK government has published the first draft of its proposed policy on the use of open source software and is seeking comments from the public. The policy essentially seeks to increase the use of open source software at all levels of government and public sector IT provision. Behind this broader acceptance of open source solutions lies a desire to ensure interoperability and the support of open standards and specifications. The formulation of the policy was driven by the European Union's *eEurope – An Information Society for all* strategy, which centered on plan that

"during 2001 the European Commission and Member States will promote the use of open source software in the public sector and e-government"

Source: The Register

Vendor Support for Open Source

Vendor support for a particular platform is a very good indicator of the potential of that platform. After all vendors are not charities and their revenues and profits are the goal of all businesses. Many vendors today either directly (*by making their software Open Source*) or indirectly (*by using or selling solutions which run on Open Source platforms or components*) support Open Source. The list of vendors is impressive and surprising:

IBM	Apple	Sun Microsystems	HP/Compaq	Veritas
Oracle	SAP	Checkpoint	Microsoft	Computer Associates

IBM

The Linux platform is aggressively supported on IBM's Intel and PowerPC based servers as well as S/390 mainframes and many applications and tools have been ported such as WebSphere, Tivoli, DB2, and Lotus Notes. IBM's Linux Technology Center is also developing Open Source software such as JFS (*Journalled File System*) for Linux and other Java based projects.

Apple Computer

Apple's latest operating system MacOS X is built upon the Darwin operating system with the Aqua user interface and associated APIs. Darwin is a derivative of FreeBSD (*FreeBSD is the operating system which powers some of Yahoo's and Hotmail's servers*) and the Carnegie Mellon's MACH 3.0 micro kernel. Apple has Open Sourced Darwin under Apple's APSL (*Apple Public Source License*).

Other Apple Open Source products include Darwin Streaming Server, Common Data Security Architecture (*CDSA*), OpenPlay and Apple Open Directory. Darwin Streaming Server is currently the most robust and complete Open

Sourced media streaming products available.

Sun Microsystems

Sun's biggest contribution to the Open Source community is OpenOffice (*an office application suite much like and compatible with Microsoft's Office suite*) even though Sun sells a proprietary version, StarOffice 6.0. In addition, Sun new Linux server line increases Sun's support of Linux in addition to its Cobalt product offerings. Sun's newest Solaris 9 operating system also comes with many Open Source components such as the Gnome Desktop Environment, Apache web server and OpenSSH. Sun is also participating in the development of Gnome, Mozilla web browser, Jakarta Tomcat application server and others.

Microsoft

Even Microsoft, the most vocal opponent of Open Source licensing (*Specifically GPL*), uses Open Source software in its products. Microsoft Windows 2000 and later use the TCP/IP networking code from FreeBSD which replaces its own TCP/IP code used in previous versions of Windows.

The Open Source Advantage

Although cost reduction is one of the primary reasons to go with an Open Source solution, there are many other compelling reasons for end users as well as software developers.

Advantages for End Users

With most Open Source software, users have direct access to designers / developers for support, bug fixes and new features. Unlike proprietary software, there is no “customer service” to deal with for Open Source software. Feedback usually goes direct to the author or contributors of the software. In this way, frequently requested features can be quickly added to new releases and bug fixes can be available in a short time. Unlike proprietary software vendors to whom bug fixing is a cost and a necessary evil, Open Source authors motivations are not financial and tend to respond faster to bugs found. Additionally vendors are always under pressure to speed up development

cycles to get the next “feature packed” version out to the market. It is common practice for new products to be prematurely released (*and hence tend to be buggy and incompletely tested*) in reaction to a competitor’s product announcement or release. The “release-early-and-fix-later” practice seems to be becoming routine. Although Open Source developers are not entirely free of such pressure (*requests from users*), they are not “forced” to release revisions prematurely due to competitive and marketing pressures.

Open Source software tends to be platform independent and generally adopt open standards, in general such software is easily ported run on any other standards based platform such as POSIX compliant systems. This can mean lower hardware and software costs and greater choices are available.

Proprietary software has a tendency to grow in size and minimum required resources, demanding faster computers and more disk storage and memory. This is due to the need to have more features to sell the next version or more features than a competitive product. Ultimately this results in bloated software with more features than most users will ever need and this fuels the never ending computer upgrade cycle with systems becoming obsolete within 18 months of purchase even it still running perfectly. Open Source software has no such pressure to sell more and more features and can frequently run well on lower specification hardware and thus allows the reuse of “obsolete” hardware.

Under some circumstances such as deployment in a military or high security environment it is always advisable to ensure that software is bug free and secure, with no “trapdoors”, “trojans” or “spyware”. Unfortunately the only way to confirm this is by performing a source code audit, which is only possible if the source code is available. No software vendor provides source code under all but the most extreme circumstances. Programmers often include unauthorised (*in some companies, this is a termination of employment offense*) code in programs in the form of “easter eggs” which usually remain undetected until the software is installed in a user's computer. Although “easter eggs” are generally harmless (*displaying pictures of the developers or plays music etc*), it does show what can escape a software vendor's checks. The Open Source advantage of having source code freely available and open would be ideal in such environments.

Advantages for Developers

The biggest single advantage of Open Source to a developer is source code availability. This allows programmers to look at how others have approached various programming problems and how various functions are implemented and this provides an excellent learning experience.

Derived solutions or “greater works” are solutions which build upon existing software. This “software building block” approach is the key to leveraging the greatest advantage of Open Source software, reusable software. This allows developers the freedom and more time to create new innovative solutions without having to “reinvent the wheel”.

Derived solutions may be as simple as improvements to a software or an integration with other software components to form a total solution. Software improvements are impossible without access to source code and integration is not usually possible with proprietary software unless vendors have agreed to adopt specific interfaces and standards to facilitate integration. In all cases of in-house deployment, there are no serious licensing issues. However, if redistribution or Internet exposure is intended licensing terms must be adhered to. For example, if improvements or modifications to a GPL licensed software are made, the derived product must also be GPL licensed. This means license fees cannot be charged for derivatives of GPL licensed software. Although it appears that Red Hat is charging for its Linux distribution, the fee paid by the “purchaser” is for the packaging, manuals, media, technical support, distribution costs etc and not paying license fees for the software itself. This is not the case with BSD type licenses where license fees can be charged for a product or derivatives as long as the “seller” does not claim to have written the original software and credits the original author(s). There is a long standing argument as to which licensing scheme is more “free”. BSD advocates claim the BSD license is more “free” (*as in freedom*) due to its nature.

The Open Source development model leverages community skills and expertise. Consider Apple's Darwin (*the core of MacOS X based on FreeBSD and the Mach micro kernel*) operating system as an example. Apple could have taken the approach of keeping Darwin source code proprietary (*which is possible since FreeBSD's licensing is a BSD type license*). This would have meant that Apple would have to maintain a large development team which would be

under pressure to churn out new improved versions while finding and fixing bugs, typical of proprietary software vendors. Apple instead released Darwin under Apple's Open Source license, Apple Public Source License (*APSL*). Darwin development is now controlled by the Open Source community with Apple contributing to the development effort. Bug detection and fixing is also now handled by the community. This allows a much greater sharing of knowledge by enabling more people to contribute to improving Darwin. Apple's commitment to Open Source has encouraged the community and this has resulted in the contribution by many, not only to Darwin development itself but also to porting applications to MacOS X.

Software portability is an important advantage to any developer. If software can be developed once on a selected platform and ported easily to other platforms, the potential market for that product is wider. Similarly in-house developed applications for internal use can be ported easily to other hardware platforms which will enable software reuse and reduces vendor dependence. It is therefore important to have some standard or guideline for applications and operating systems to enhance portability between platforms.

The IEEE PASC (*Portable Application Standards Committee*) is the caretaker of the POSIX (*Portable Operating System Interface*) standard for application and programmer portability. Almost all current Unix systems (*including Open Source versions*) are POSIX compliant (*which is a US Government requirement for procurement*) and this is a major improvement to portability between POSIX compliant systems.

Although Microsoft Windows NT / 2000 is compliant to IEEE Std 1003.1-1990 – POSIX.1 (*they are not compliant with the later 1003.2-1992, POSIX.2 or the latest revision to POSIX.1 1003.1-2001*), Windows based applications tend to use Windows APIs and rarely use POSIX APIs. This minimal POSIX compliance was required to enable Windows NT/2000 solutions to be proposed for US Government contracts. Open Source operating systems such as FreeBSD and GNU/Linux on the other hand are working towards as high a level of compliance as possible with the latest POSIX 1003.1-2001 standard. Additionally portability between Unix platforms is also improved by adherence to the Open Group's Single Unix Specification, Version 3, the core of which is POSIX 1003.1-2001. This means porting from Windows to Unix platforms very difficult due to the lack of Windows APIs on Unix platforms (*on Intel hardware platforms the Wine project, www.winehq.org, is working to provide Open Source Windows APIs and libraries to allow Windows software to run*). However, conversely, porting from Unix

platforms to Windows is relatively easier, due to GNU development tools and RedHat's CygWin, a Unix and POSIX compliant API for Windows and both are Open Source licensed under GPL.

Although standards like POSIX and The Single Unix Specification do so much to improve the portability of software, which ultimately benefits the purchaser of software, adoption of such standards is far from ideal. The United States government is one of the largest entities which specifies POSIX compliance in its procurement requirements. This was instrumental in “encouraging” entities like Microsoft to seek compliance. Unfortunately the compliance level was the minimum needed to meet the procurement guidelines. It is therefore clear that standards adoption has to be driven by large entities such as governments.

The Open Source Dilemma

Given the many advantages of Open Source solutions, there are still many who are wary of deploying such solutions. Many misconceptions are largely due to lack of information or real knowledge about what Open Source is really about. Another reason is FUD (*Fear Uncertainty and Doubt*) campaigns by those opposed to the Open Source concept.

Common Misconceptions About Open Source

Open Source software is sometimes considered insecure because source code can be searched for bugs to exploit. This is stated by proponents of “security by obscurity”. However proprietary software has as many if not more exploits, which proves that even if source code is not available, exploits will be found. In fact, many Open Source developers such as the FreeBSD Foundation have regular code audits to look for poorly written code, bugs and potential security issues. These code audits will simply not be possible without free access to the source code. Most proprietary software developers are under pressure of release deadlines and do not include code audits as part of their pre-release procedures and instead rely on end user functionality testing which will not reveal flaws in the code. Without the pressures of deadlines, Open Source developers can afford to take the extra time to properly audit and re-audit code. In addition to the developers the community at large also contributes by reviewing the code (*peer reviews*). With so many “eyes” examining the source code many potential problems have been found and fixed before exploits have been discovered and distributed. Code audits are time consuming and it is likely that vendors of proprietary software would rather use their limited resources to develop new features (*which are revenue generating*) than search existing code for bugs (*additional cost*). It is this “we will fix it when it breaks” policy by software vendors, which results in the proliferation of viruses and trojans on the Internet today. In many cases even these fixes are delayed for weeks or months and sometimes never addressed.

Another misconception is that Open Source code is inferior because something given away free cannot be better than something paid for. Although not all Open Source software is well written, the same can be said for proprietary software. In fact many developers of Open Source software work for proprietary software vendors. Programmer skills are therefore not likely to play a role in determining if proprietary software is better than Open Source. The

argument that vendors have more resources to produce better software is also false. Vendor resources are usually limited to keep costs low and the focus is on adding features to the software to be sold in the next release. The keys to quality software, are code audits and peer reviews, this can only be efficient if enough people have access to source code.

Another misconception is that Open Source solutions are unsupported because there are no vendors to turn to for support. Although it is true that not all Open Source software has a user support organisation, support usually comes from the author and community (*user mailing lists and newsgroups*) which is often more effective than many vendor's helpdesks. In addition, besides Apple, there are commercial organisations such as Red Hat, IBM, HP/Compaq and Sun which sell and support Open Source solutions on their equipment. It is however true that to gain maximum benefit from Open Source, an internal team of dedicated people is important. With less well-known Open Source solutions the implementor has to determine fitness for a particular purpose which means an experienced team is essential. Meanwhile, as the internal team is being trained and building experience, there are many companies to which support of Open Source solutions can be outsourced to. Although Open Source licenses are free, Open Source is not "Zero Cost". Like any proprietary software, there needs to be an implementation team to get it working and a maintenance team to keep it working.

Another frequent misconception is that Open Source solutions are suitable only for small scale deployments as they are unscalable. This has been proven to be completely false. Scalability is a function of the overall design of the of the systems and network architecture and not the origin of the software. Some of the largest E-commerce and portal sites in the world run on Open Source software solutions such as Amazon.com, Yahoo and Hotmail.

Other Reasons Slowing the Adoption of Open Source Solutions

One of the primary reasons why many organizations are slow to adopt Open Source solutions is that although individual software solutions are well supported by their respective developers and user groups, there is a lack of support for integrated solutions, that is solutions which comprise of a combination of many individual components which make up a total solution. Community support may be available for popular combinations but unusual solutions need a high level of internal skills to build and maintain. It is the lack of such skills internally and in the market at the moment, which makes organizations wary of Open Source solutions.

The major reason for the lack of skills in the market is the current state of the education sector. Due to the great

demand for IT trained people, many IT related courses are conducted by many colleges and other institutes of higher education. However, the lack of suitably skilled teaching staff leads to poorly designed courses, which do not teach the principles of IT or teach students analytical skills. Instead many teach courses on how to use proprietary software packages. Programming is taught using proprietary software development products and not the principles of programming, algorithms and multiple languages. To make matters worse, many such courses do not have any accreditation. The result is poorly trained graduates without any foundation in the principles of their craft and little or no analytical ability. It is not surprising that many employers now days require job applicants to have additional professional qualifications. These graduates also go into industry with a preference for the proprietary software they are familiar with.

Another issue with Open Source operating systems is a lack of consistent look-and-feel. Open Source OSes primarily depend on XFree86 (*the Open Sourced version of X Windows*) to provide GUI functionality. Unlike GUIs such as Microsoft's Windows and Apple's Aqua with integrated window managers and environment toolkits (*which provide a consistent look-and-feel*), XFree86 depends on environment toolkits and window managers such as KDE, GNOME, Windowmaker and many others. More than 20 window managers and variants are available to date, such is the freedom of choice available. However, this large number of choices does lead to two problems, user confusion and lack of consistency. This lack of consistency causes problems when installing software which utilise libraries from different window managers, such as QT libraries for KDE applications and Gnomelibs for Gnome applications. A UK government funded analysis of the use of Open Source software, "Analysis of the Impact of Open Source Software", concluded in October 2001 that although Open Source solutions showed great promise in the server environment, they did not consider that there was a viable alternative to Microsoft Windows on the desktop. However they did recognize that this could change and recommended a re-evaluation within a year.

The lack of traditional software distribution channels also limits the adoption of Open Source solutions. Many solutions are only available in source code format. It is up to the user to download and compile the source code into an executable format. This puts these solutions out of reach of the vast majority of users who do not have the expertise or inclination to do this. Although major operating systems are available off the shelf in shrink-wrapped packages, the vast majority are source code only. A case in point is the Open Office productivity suite. It is only available in precompiled binary executable format for a limited number of platforms and these can only be

downloaded. Those without broadband access have difficulty obtaining the 200 Mbyte package. For the rest, it means downloading the 150 Mbyte source code package. Following this are the hours needed to compile the package (*approximately 9 hours on a 1Ghz Pentium 3 system*).

Limited hardware support is another weak area in Open Source operating systems since many hardware vendors are unwilling to publish information to interfaces to their products or spend resources creating drivers for other platforms. This severely limits driver availability for many products. A case in point is the lack of direct support for many accelerated graphics products for X-windows which have to use the un-accelerated generic VESA SVGA drivers. Other products with limited driver support include notebook computers, printers, scanners and digital media products such as cameras and video cameras.

Issues Specific to Linux as a Special Case

Linux is by far the most well known Open Source operating system at the moment. Many even incorrectly regard Linux as being synonymous with Open Source due to its wide coverage in the media and the strength of the Linux movement. Linux is in fact the name of the kernel (*the core of an operating system which handles CPU and memory resources*) of the GNU operating system by the Free Software Foundation called GNU/Linux. GNU/Linux has many variants, called distributions, which consists of GNU/Linux and customized components and libraries specific to the various distributions such as Red Hat, SuSE, Debian, Mandrake and many others. This great diversity of GNU/Linux variants is both its greatest strength and weakness. Incompatible libraries and documentation often creates problems for beginners when compiling software written for a different distribution. In addition some distributions have different and incompatible package managers for precompiled software such as Red Hat's RPM and Debian's dpkg. Recently with version 7.0 of Red Hat's distribution, the introduction of a binary executable format which is incompatible with other Linux distributions created a stir in the community. It is this corporate profit driven need to distinguish some distributions from the rest which is fracturing the movement and so weakens the overall Linux case.

Closing

The case for Open Source grows stronger daily and no one can deny that the concept is a serious threat to traditional software development and licensing paradigms. Even if cost advantages are ignored there are many important reasons to at least consider Open Source solutions as an option. Openness is by far the most important factor. The 1980's buzz phrase "Open Systems" has truly been reincarnated as Open Source. The goal of Open Systems was to ensure interoperability, compatibility and portability between systems and operating systems. Although the Open Source concept did not have these as primary goals, it created the environment which encouraged the adoption of standards which in turn led to improved interoperability, compatibility and portability. Without commercial pressures to out-do the competition, Open Source developers cooperated with each other, shared code and helped each other get their software running on each other's systems, regardless of operating system or hardware.

Cost is a major motivating factor for businesses, more so in today's economic situations, which makes Open Source solutions very attractive. However, it is important to realise that free licensing does not mean zero cost. Planning, implementation and maintenance. Open Source solutions offer free licences but unless in-house technical expertise is available, it may be harder to realise the maximum potential for cost reduction. As there is no vendor to turn to, in-house staff have to manage the process of identifying solutions, deploying and integrating them with other solutions and then maintaining the solution. This means in-house staff must have the required expertise and such expertise is not by any means cheap. As an alternative, outsourcing the support of Open Source solutions is certainly viable and may be less expensive than retaining in-house expertise on a full-time basis. The real savings will come when the business grows but costs remain stable, without the ever escalating per-user, or per-data record, or per CPU licensing charges associated with proprietary solutions.

Appendix 1: Sample of Open Source Solutions

Internet Server Applications

Apache web server – the world's most popular web server with about 60% of all web sites

OpenLDAP directory server

Sendmail (*most popular MTA, message transfer agent, handling 80% of the world's email traffic*),

Qmail SMTP MTA POP3 server

Postfix SMTP MTA

Cyrus POP3 IMAP4 server

Exim SMTP MTA

Qpopper POP3 server

BIND – almost all the world's DNS servers run BIND or derivatives

Darwin Streaming Server – video streaming server

Database

MySQL database server

PostgreSQL database server

Application Servers

Zope application server

Jboss and JOnAS Java 2 Enterprise Edition (*J2EE*) application servers

Security Software

Nessus security vulnerability scanner - Detected more vulnerabilities than any proprietary scanner (*Network Computing Magazine January 2001*)

Snort Intrusion Detection - Rated 3rd in a review after Enterasys Dragon and Cisco IDS. RealSecure IDS was 4th (*Network Computing Magazine August 2001*)

T-REX, Dante, IPFW – Firewalls

NMAP – Port scanner

OpenSSL – Secure Sockets Layer Libraries

OpenSSH – Secure Shell

Network Management Tools

Ethereal – network protocol analysis tool

Nagios – network and services monitoring system

MRTG – network traffic monitor

File and Print Server Applications

Samba SMB domain master, file and print server for Windows and SMB/CIFS clients

CUPS – platform independent print solution

Ghostscript – postscript interpreter

Operating Systems

Linux – kernel for the GNU operating system (Red Hat, Debian, SuSE, Mandrake etc)

FreeBSD, OpenBSD, NetBSD, Darwin – BSD 4.4 Lite based operating systems

Client Software

Mozilla – standards based web browser

Konquerer – standards based web browser

Sylpheed – email client

Stuphead – email client

Evolution – email client

Office Applications

OpenOffice – office application suite

Koffice – office application suite

ABI Word – Word Processor

Gnumeric - Spreadsheet

High Performance Cluster Computing Software

BLAST – molecular sequence search tools

MPICH – portable implementation of MPI message passing libraries

OSCAR - (Open Source Cluster Application Resource) is a snapshot of the best known methods for building, programming, and using clusters

Desktops

XFree86 – X11 X Windows network windowing implementation

Windowmaker – X11 Window Manager

KDE – X11 GUI Environment

GNOME – X11 GUI Environment

CAD and Graphics

Xcircuit – schematic capture application

The GIMP – Photo/Image editing solution

ImageMagick – Image manipulation tools

QCAD – 2D CAD program

SPICE – circuit simulation software

Software Development tools

GCC – GNU Compiler Collection

Perl – scripting language

Appendix 2: Sample of Open Source Organisations and Resources

(Some of the descriptions below are sourced from the web sites of each organisation and remains the copyright of the respective owners)

The Open Source Initiative

The OSI is a non-profit corporation dedicated to managing and promoting the Open Source Definition for the good of the community, specifically through the OSI Certified Open Source Software certification mark and program.

www.opensource.org

The Free Software Foundation (FSF)

The FSF is the principal organisational sponsor of the GNU Project. FSF's mission is to preserve, protect and promote the freedom to use, study, copy, modify, and redistribute computer software, and to defend the rights of Free Software users.

www.gnu.org

Apache Software Foundation

The Apache Software Foundation provides support for the Apache community of open-source software projects. The Apache projects are characterized by a collaborative, consensus based development process, an open and pragmatic software license, and a desire to create high quality software that leads the way in its field.

www.apache.org

FreeBSD Foundation

The home of the FreeBSD operating system. The goal of the FreeBSD Project is to provide software that may be used for any purpose and without strings attached.

www.freebsd.org

The Linux Kernel Archives

Primary Repository for various versions of the Linux Kernel source code.

www.kernel.org

Linux Online

Linux Online's web site is designed to act as a central source of Linux information and as a voice for the promotion and advocacy of the Linux operating system. Their main goal is to inform the public about every company, project and group that uses the Linux operating system and to report on the hard work of countless developers, programmers and individuals who strive everyday to improve on the Linux offerings in the marketplace.

www.linux.org

OSCAR (Open Source Cluster Application Resource) Project

OSCAR is a snapshot of the best known methods for building, programming, and using clusters. It consists of a fully integrated and easy to install software bundle designed for high performance cluster computing.

oscar.sourceforge.net

The Open Channel Foundation

A portal for open source software for CAD, CAM, CAE which provides solutions for scientific, engineering, industry and technology

www.openchannelfoundation.org

The OpenBSD Project

The OpenBSD project produces a FREE, multi-platform 4.4BSD-based UNIX-like operating system. OpenBSD efforts emphasize portability, standardization, correctness, proactive security and integrated cryptography.

www.openbsd.org

The Internet Software Consortium

The Internet Software Consortium (ISC) is a not-for-profit corporation dedicated to developing and maintaining production quality Open Source reference implementations of core Internet protocols.

www.isc.org