Middlesex University Research Repository

An open access repository of

Middlesex University research

http://eprints.mdx.ac.uk

OPEN SOURCE SOFTWARE IN HIGHER EDUCATION

Alex Birchall and Alan Hopkinson
Middlesex University, Learning Resources
London, UK

Abstract
A description of Open Source software is given and reasons are given for its appropriateness in Higher Education. Methodologies are proposed for ensuring its greater take-up in Higher Education – should Higher Education actively develop in this direction? - and some recommendations are given on how the sector can secure the use of Open Source.

1. What is Open Source?

Since the beginning of the computing era, there has been a proportion of system developers usually looked upon as enthusiasts first and perhaps professionals second who have been keen to develop systems in cooperation with others, usually a very loose form of cooperation. There was the shareware movement where programmers would work on a programme and make it available to the world through various 'networks' which in those days meant providing floppy disks in return for voluntary financial contributions. When the internet appeared on the scene, it all but killed off shareware because it became too easy to distribute this software: no one was prepared any longer to pay for it as it is so easy to download from the internet and the person acquiring the software usually never gets round to the more difficult task of paying for it.

Open Source Software is the successor of this movement in the internet era. There is a worldwide "movement" underway related to open source software development; consequently the term has acquired many different meanings depending on one's perspective.

Strictly speaking, "open source" means that the software source code is:
1) made available for others to use, view, and modify; and
2) that it may be redistributed by anyone for free, without royalties or licensing fees to the software owner.

In contrast, source code from purchased software packages generally is not distributed or made fully accessible to anyone and in many cases users are prohibited from copying or redistributing the compiled software.

The term "open source" is however commonly used to denote a particular model of software development. This is the decentralized approach taken by well-known and established projects such as Linux and Apache, in which a world-wide community of programmers contribute to the development and ongoing maintenance of the programs. There is a wide continuum of motives for those who contribute to open source software.

At one end of the spectrum are unpaid volunteers motivated by the intrinsic rewards of solving an interesting problem, by the prestige one can gain within the programming community or the user community, and by the desire to create something for one's own use.

Second, there are programmers assigned to add locally needed features and contribute to the improvement of open source software by organizations that have made a business decision to adopt it.

Thirdly, there are companies that choose to enhance opens source software because they sell services related to that software. This third case includes even companies like IBM, which develops open source software in a centralized, controlled environment. They can earn their living by providing maintenance and other value-added services associated with the software.

The two meanings of 'open source' are difficult to separate completely and are often confused.

2. Why Open Source in Higher Education

2.1 Availability of Programmers

Open Source is attractive to the Higher Education sector (HE) where there are computer programmers available with many more roles than are found in commerce or industry.

There are students who need to do programming projects, academics who want experience, interns in computing centres who are on courses from that or other universities, recent graduates needing more experience before joining the commercial world. So HE is well placed to find the programmers it needs.

2.2 Origins of Open Source Software

Many of the best known commercial products used by HE began in universities. Blackboard, the VLE software, was originally developed by and for the US University of Cornell. Sun Microcomputers Operating System was originally Berkeley's version of UNIX. Cisco's network systems were built on Stanford's networking software. Most library automation systems began in a university and were developed by shared activity. The best example was DOBIS/LIBIS (University of Dortmund with support from IBM on whose minicomputers it ran), DYNIX and VTLS (Virginia Tech Library System) TALIS, formerly BLCMP which was Birmingham Libraries Cooperative Mechanization Project. The UK Heritage system was developed for University College Oxford. Other systems which no longer exist originated in the HE sector and failed later when in the hands of the commercial sector such as LIBERTAS which began as South West Academic Libraries Cooperative Automation project.

In the digital libraries field, Greenstone has been developed by Waikato University and made available through various channels including through the sponsorship of UNESCO. In the repositories field, Southampton University has developed eprints and DSpace, developed jointly by MIT and Hewlett Packard are both Open Source products.

Open Source is not the same as free! Software such as the members of the CDS/ISIS family are not open source. CDS/ISIS for DOS and Windows is freely available but the Source software which developers need to adapt to tailor software to their own needs is not available as much of the software is only available in the compiled version. Software may be made freely available but if the source is not available the software is not Open Source. UNESCO has a policy to foster Open Source software so it is ironic that one of their best known packages is not Open Source and the only way it could be would be if it were rewritten from scratch. BIREME in Brazil are working with UNESCO to produce an Open Source version.

2.3 Current Open Source in Higher Education

Formerly, universities developing software for themselves which they felt had value for others tended to sell it to a company or at least to set up a company at arm's length to support it. Now through Open Source there is a model for cooperative and shared development. To meet the specific needs of HE a number of packages have been developed through Open Source

Over the past ten years software has assumed a critical role in higher education. In the early 1990s students and faculty were just beginning to use email in earnest. Today, software is woven into almost every aspect of HE, from enterprise resource planning systems used to administer universities, to the distributed research efforts faculty routinely engage in, to systems supporting Virtual Learning Environments (VLEs) and online courses used for teaching, to institutional repositories and publishing tools used to disseminate research. Campus networks must support local and remote users, provide a high level of security, and meet a broader array of software needs than most large corporations. These include campus portals, student portfolios, personal information managers, and peer-to-peer file sharing tailored for the academic community. Sophisticated software applications are also playing an increasingly prominent role in some areas of the sciences and social

sciences. To meet these demands, many universities have built up large IT departments, and in some cases develop their own software to meet these specific needs.

Over a dozen open source projects have been launched within the higher education community to address many of these needs.

The oldest of these is uPortal, which is a free, sharable portal-enabling software which allows academic institutions to provide abridged and customized versions of their campus Websites. uPortal compares well to commercial alternatives and has been adopted by hundreds of institutions worldwide.

major project underway is Sakai, a community-based software development effort to design, build and deploy a new Collaboration and Learning Environment (CLE) for HE. Sakai is led by staff at the University of Michigan, Indiana University, MIT and Stanford.

2.4 Open Source in the HE Sector

The commercial world has leaders, usually though not always the largest corporations, with the funding to put the necessary resources into providing exactly what the sector needs. The lesser institutions can ride on the backs of those. However programs developed for their administration often are not suitable for use in the academic sector. Human Resource management programs like Peoplesoft are developed for much more structured environments. Oracle Financials similarly.

Open source is increasingly viewed as a viable alternative to proprietary software and is having a substantial impact on the way commercial organizations do business. Linux, the best known Open Source operating system has a 28% share of operating systems on enterprise server computers, and Apache web server software is used by 67% of all websites. MySQL, the best known open source data base management system has over six million active installations worldwide. Major corporations such as IBM, Hewlett-Packard, Sun Microsystems and Intel have become supporters of Linux and hardware suppliers sell machines loaded with Linux software. IBM, Oracle, SAP, BEA, Veritas and Intel, as well as many smaller firms such as Red Hat, have created programming groups assigned to work on open source software and provide support services.

There is also enormous interest worldwide among developers. The SourceForge.net website lists over 100,000 open source projects in progress covering almost every kind of software and has more than 1.1 million registered users.

Many people have been suspicious of Open Source. Shirish Netke of Aztec Software writing in sandhill.com a website for software executives about business strategy reminds us that "open source is not about free software. It is really about a new business model for software services." [1]

This is becoming an increasingly widely shared opinion in the software development industry.

Something which like shareware began as something done by and for enthusiasts has taken off commercially. Why has this happened? It must be that unlike shareware it offers a sustainable business model.

Additionally, open source software can offer greater flexibility for the user. Users can choose to avoid becoming locked in to software provided by a particular vendor, they are not compelled to upgrade to the same extent. Vendors who support open source software have an incentive to offer a better service because switching costs for customers are much lower than with proprietary software. Open source code can be customized by users, and improvements can be shared among the user community. Users can choose the features that are important for their own institutions. And open source code is usually based on open standards that foster integration and interoperability. An indirect benefit for users is that, where a credible open source alternative exists, they may gain leverage in negotiations with vendors and can put pressure on them to provide greater flexibility and transparency.

Of course one must not forget that there can be disadvantages in the open source model.

First, since the software itself is distributed for free, funding may not be available to support all the features users want when they want them. But this is true of formally purchased commercial systems.

Second, as most early successful open source software has been developed by technical people for technical people; these projects (particularly those following the community development model) did not tend to invite the user-centric design approach that is desirable for end-user applications, although it must be pointed out that recent new open source applications such as Firefox (an internet browser) appear to be overcoming this hurdle.

Finally, there is often no legal entity responsible for indemnifying users, who may be exposed to copyright infringement lawsuits. The fact that an established company such as Lloyd's of London is about to launch an open source insurance product shows that this risk is a cause of concern for many – and also illustrates how mainstream this movement has become.

Does Open Source save money? In certain areas it certainly does. Apache is free and causes very few problems. Firefox is a free web client and that can be installed and causes few problems. If there are problems there is a community of people out there ready and willing to help. Of course no one has to use Firefox and its alternative, Microsoft's Internet Explorer is available to everyone who has obtained Microsoft's Operating System. However in the future some of these OS products may cease to be developed and users would have to return to Firefox. However, any product, commercial or OS has a risk of being discontinued, even if the product is successful. This happened recently in the case of the Horizon software which we have at Middlesex University.

3 Should higher education develop open source software to meet its own needs?

Champions of using open source in HE say that academic institutions have unique needs that are not well served by commercial vendors because the sector is too small to support a robust competitive market. Higher education accounts for a relatively small share of the IT industry.

The relatively small size of this market does not attract the level of competition some would like to see in areas of interest to higher education institutions such as VLEs and financial systems targeted to the community.

As a result, one of the most often-heard complaints is that available software options do not have the features required or do not allow for cost-effective customization. In the library field, the three or four options for integrated library systems, all of which are proprietary, are expensive, contain bugs, developments are slow, and in general they do not allow customers to tailor them to their own specific needs. Additionally they are not that popular with users. As far as open source alternatives are concerned there are Koha, Emilda and Evergreen. Evergreen is in use in largish libraries but does not have acquisitions; Emilda is a new package produced in Finland and Koha has only been used by a very few academic libraries worldwide. Certainly the community could if it developed its own software put pressure on commercial vendors to make their products more open, more flexible and more easily adaptable.

Another common concern of many administrators these days is that academic institutions are too much in the hands of commercial vendors. The acquisition of PeopleSoft by Oracle in late 2004 is probably the most noteworthy example. After spending huge sums installing PeopleSoft, many university administrators fear that Oracle would stop supporting PeopleSoft's financial systems in an effort to force their institutions to switch to Oracle products. They were also concerned that the consolidation of the two leading providers of this type of software to higher education would result in higher prices.

A second commonly cited example of higher education's dependence on proprietary software is Blackboard. Over the past seven years Blackboard has put itself in a position where it has almost a 50% share of the VLE system market. Blackboard's power over its customers has been an ongoing source of frustration, as institutions believe they are forced to buy unwanted features, and expensive

customization efforts must be repeated when Blackboard announces that it will no longer support an earlier version of its software.

The fact that the higher education market is of relatively modest size also raises the question of whether the academic community is large enough to support its own software. Projects like Linux and Apache have been built by huge global communities, but then uPortal has been developed with minimal resources and now runs on several hundred campuses and is a well established and valued option. The Sakai project alone concentrated $6.8 million invested over two years by the Mellon foundation and the four partner universities who developed it.

Blackboard, the leading commercial courseware management system provider, set aside $14 million last year for research and development on its product. With proper coordination and commitment, could the community development model rise to sustain itself at such a scale? It would depend in part on the extent to which these projects could attract contributions from an international community and the extent to which member institutions supported their IT staffs' participation in such a commons-based production process.

4. What approach should higher education take to securing the future of open source projects?

In the UK we have JISC, the Joint Information Systems Committee joint between the library and computing communities in HE.

They have set up Open Source Technology Watch http://www.oss-watch.ac.uk/ [2]. In other countries there is no such body and in the United States it is left to the voluntary sector to work with communities which in most cases are supported much more by charitable giving from a variety of sources than are most European HE institutions.

However, the Mellon Foundation is supporting Open Source. Carnegie Mellon West University has set up COSI which will focus on methods and practices for developing, adopting, managing, and integrating systems that incorporate open source technology. COSI will establish a laboratory, library, and community website that can serve as a testbed and experience factory for open source software, as well as a site for student and faculty projects.

So Higher Education institutions need to keep in touch with the developments taking place as reported in these websites.

References:

1. Netke, Shirish. "The Real Open Source Revolution" SandHill.com, June 6, 2005. http://www.sandhill.com/opinion/editorial.php?id=30
2. JISC *OSS Watch Open Source Advisory Software Service,* 2007. http://www.oss-watch.ac.uk/
3. Centre for Open Source Innovation. http://cosi.west.cmu.edu/