Project Document

Report on Open Source Learning Object Repository Systems

Synopsis:

This report examines developments and issues related to the use of Open Source Software/Free Software (OSS/FS) in relation to the JORUM learning object repository (LOR) service. Evidence is cited to show that using OSS/FS is becoming more acceptable, especially within the education sector, and international governments are developing policies about how to use this software development model. Substantial consideration is given to the issues raised in procuring OSS/FS, especially to those of support and maintenance. The importance of building up a supportive community around OSS/FS projects is highlighted.

After reviewing current OSS/FS LOR we could find no such systems in service at present. However, the current interest in developing OSS/FS institutional repositories (e.g. DSpace) for various types of scholarly material suggests that the relevant technology exists. In particular, the "toolkit" approach, involving the development of OSS/FS components, may lead to fully featured OSS/FS LOR in the future. The importance of Open Standards and Specifications in the digital repository field is emphasised so that interoperability is facilitated; this is compatible with the e-Learning Framework and the growing adoption of Web Services in a distributed e-Learning technical environment.

Authors: The JORUM Team

Date: November 2005

Version: Final

Status: Final

Authorised: Signed-off by JISC

Version Changes to this File

REVISION	DATE	CHANGE	DETAILS
v1.0	11/07/05		
v1.1	28/7/05	JC	Edits and corrections
V2.0	29/7/05	JC	Final edits. Send to JISC
FINAL	23/11/05	AD	To incorporate comments made by SE

Related Documents

RELATED DOCUMENT	DETAILS
JORUM Report: Monitor Technical Frameworks and Infrastructures	Due 2/9/05
JORUM Report: Repository Systems Research Watch	Due 2/9/05

Contents (Sections)

Copyright Notices

[1] GNU Free Documentation License available from: http://www.gnu.org/copyleft/fdl.html

1. Summary and Conclusions

1.1 Summary

Given the length of this report and the amount of detail it provides, it is helpful to provide a summary of the main points given in each of the foregoing sections.

The momentum in favour of giving OSS/FS proper consideration alongside more conventional proprietary software solutions is growing. Internationally, many governments are using OSS/FS for public services, including the education sector. The UK government now has a policy that basically requires their overall software procurement to be good value for the public money invested, as well as being in conformance to open standards and specifications applicable to that type of software. The policy states that OSS/FS needs to be given a fair hearing when such procurement decisions are being made by publicly-funded bodies. The UK government requires that certain features be mandatory:

- The software conforms to open standards/specifications
- That lock-in to proprietary IT products and services will be avoided

The guidelines for the UK JISC services follow government policy including providing unbiased advice about OSS/FS and proprietary software. The Oxford-based OSS Watch serves this purpose.

It is important to be clear about the meaning of OSS/FS and the reasons for choosing this type of software development model. Although governments clearly include value for money as a major consideration, this report suggests that this should not be the only criteria adopted when choosing OSS/FS. The main feature of this development model is that the code should be open to inspection by the widest possible community of developers and, hopefully, be capable of rapid development, bug-fixing and adhere to Open Standards as much as possible.

From the section 3.3 discussion of Organisational Developments in the OSS/FS world it seems clear that community support for users of OSS/FS is growing very quickly. Building supportive communities is a major interest of some recent OSS/FS conferences (e.g. OSS-Watch at Edinburgh, 2005).

From section 3.4 it is also clear that OSS/FS is now being widely used in various international governments and especially in their education sectors. The recent BECTA report indicates the belief that use of OSS/FS can be very cost effective for the Schools reviewed in that report. The Higher Education sector has been using OSS/FS for some time and there are now some interesting attempts to integrate OSS/FS systems, such as the Virtual Learning Environments (e.g. MOODLE) and LAMS, along with assessment tools (e.g. TOIA) using single sign-on access. Many universities are investigating the use of OSS/FS for their Institutional Repositories, with EPrints being one of the most widely used systems for such archival purposes.

Section 3 outlined the current interest in developing OSS/FS institutional repositories for various types of scholarly material. It is suggested that the relevant technology exists (and that toolkits are being developed) that may, in the near future, result in fully featured OSS/FS learning object repositories appearing. The most promising institutional repositories from JORUM's perspective seem to be those based on DSpace and Fedora since these two are well-funded at present, have fairly large user bases and the systems seem technically sophisticated enough to be modified to meet the requirements of a fully-featured LOR.

Section 5.5 suggests that OSS/FS toolkits, such as the ECL and the Apache toolkit approach using Frameworks, are promising developments since these are tried and tested methods used in existing repository systems (e.g. the Canadian eRIB). The main benefits of this approach lie

in the reusability of code; software components can be 'glued' together, using the Apache Cocoon scaleable and coherent architecture.

The international review of LOR initiatives in section 6 shows that considerable development work into OSS/FS LORs has occurred in Canada, the US, Europe and Australia. Much of the essential groundwork has involved the creation and application of Open Standards to facilitate interoperability. The development of such Standards is clearly an international matter requiring the willing collaboration of interested parties. The construction, and use, of LOR built on such standards is also international in scope. Institutional Repository systems such as DSpace, Fedora and EPrints are used outside their country of origin; some of the LOR (e.g. the Canadian COLLOR system) also appear to have a growing international acceptance.

The conclusions drawn in sections 7 and 8 are that there are no off-the-shelf OSS/FS LORs currently available for the JORUM service to use; any such LOR would probably require considerable modification and development of the underlying code, plus customisation of the user interfaces. The experiences at BCampus indicate that this fact needs to be fully appreciated. If the City College Coventry LOR were to be developed into a system that JORUM could effectively use then the above comments would also hold true. This implies considerable software development work that JISC would need to include in their plans. If the idea is to adapt the Institutional Repository systems like DSpace or Fedora into LOR this could prove to be even more difficult, if radical redesign is needed. The conclusions, however, drawn from the Connexions case study supports the argument that a successful LOR service can be developed on top of a generic content management system and bring many advantages.

1.2 Conclusions

This section will briefly discuss the question: "What would be involved if JORUM were to try to use OSS/FS for a future JORUM service?"

A valid conclusion will consist of determining whether it is technologically possible and practical (in terms of current technology, economics, and so on) in the light of previous considerations.

Based on a comparison of features of the existing repositories set against the criteria for the JORUM repository some recommendations and conclusions will be drawn about the feasibility of the particular repository, where relevant.

1.2.1 The Repositories Reviewed

From the OSS/FS repositories reviewed in section 6 it is clear that none of them completely meet the full set of JORUM's requirements at the moment. Eduplone seems to offer the closest match to the JORUM requirements although it is still in the early days of development and has a limited user group. These would be important obstacles when selected a system for a large scale repository like JORUM. However it is recognised that the use of OSS/FS is changing rapidly, and that many projects such as Connexions are building successful LORs from customised solutions.

Of the generic digital repositories, Fedora, DSpace and Plone seem to have much of the desired functionality and both show signs of continued rapid development. DLearn at Arizona and DSpace at Cambridge are investigating the issues of providing more LOR functionality. Fedora is being used as a digital assets management system supporting Elated, VUE and the ARROW partner universities.

1.2.2 Is it Technologically Possible?

It certainly seems technically feasible to make use of OSS/FS components in a future JORUM repository, if not to use OSS/FS entirely.

The toolkit approach adopted by eduSource or Apache Cocoon may be a feasible one to adopt for the JORUM project; however there would need to be considerable development work to make the individual components and tools fit together (and interoperate) successfully.

1.2.3 Is It Practical?

The use of OSS/FS repository systems (e.g. Fedora) in partnership with proprietary software developers (e.g. VTLS in the ARROW project) seem to becoming more common in some projects. Many UK and international universities already use a considerable amount of OSS/FS for their mission-critical servers. The use of DSpace, Fedora and EPrints by universities in the UK and abroad suggests that this software is very reliable and becoming increasingly more fit-for-purpose, at least for repositories of a limited scale.

1.2.4 Final Conclusions

In general, there is evidence that the OSS/FS development model is now moving into the mainstream. Many educational institutions are actively considering, or have already chosen, OSS/FS products to meet their operational goals. It is essential that any software solution is chosen for good operational reasons and this is definitely true for OSS/FS.

There are few off-the-shelf LOR systems that are currently available but there is growing interest in developing such systems; Institutional Repository systems are leading the way in their use of OSS/FS. With the emergence of systems such as Apollo and ALOHA II the trend seems to be moving towards the dynamic delivery of interoperable learning objects rather than just the storage of academic outputs.

The cleanest and most ambitious plan, would be to design a LOR from the ground up, reusing existing OSS/FS technologies to save development time. The key issues are not technical; rather there needs to be a strong enough will to make it happen. The software development skills exist but await a community that regards building an OSS/FS as a worthwhile and viable project.

1.2.5 Should JORUM use OSS/FS in the Future?

Conclusions made within this report suggest that at present there are no OSS/FS learning object repositories suitable for JORUM. The majority of the systems within this report would require large amounts of customisation to satisfy JORUM's requirements, and those that are close enough are far too immature to provide the scalability and extensibility demanded by the service in development.

There are two possible OSS/FS directions that JORUM could explore:

Option 1 - Implement OSS/FS components to support additional requirements:

> Since intraLibrary (the commercial solution procured for the JORUM Service in Development) already makes use of the OSS/FS database technology MySQL and is interoperable with OSS/FS tools such as RELOAD it seems likely that further OSS/FS components will be added to the system in the future.
>
> At present a lot of contributors use the RELOAD tool to tag resources and then bring them into JORUM. With the incorporation of the RELOAD tool into ALOHA II comes a number of exciting possibilities. The uploading of resources into JORUM could be

made as automated as possible with the use of ALOHA II and a small amount of customisation. In addition the ALOHA II tool could be configured so that users could potentially search, preview, download and repurpose learning objects through the ALOHA II tool without the need to access the JORUM repository from its front end.

Option 2 – Migrate over to a mature OSS/FS solution in the future

As user requirements expand to demand more and more features, in shorter periods of time, the OSS/FS software development model may be an efficient method of achieving these goals. Based on the contents of this report the Eduplone repository seems to provide the closest match to the JORUM requirements. The product builds on the strong attributes of the Plone application as demonstrated by the Connexions project, and incorporates the IMS specifications essential to JORUM.

When a project's requirements are so diverse, and indeed specific, like JORUM's have been, it is extremely ambitious to take on the responsibility of software development internally within the team. This, coupled with the tight timescales that the JORUM project has had to work to, makes the consideration of an OSS/FS solution during procurement very difficult. The findings of this report seem to suggest that there are an increasing number of advantages for implementing an OSS/FS repository, over selecting one from a commercial vendor. As these OSS/FS products mature and develop according to the needs of the educational sector, their implementation in large scale projects such as JORUM are likely to become more commonplace.

2. Introduction

2.1 Purpose and Scope

The Joint Information Systems Committee (JISC) of the Further and Higher Education (F/HE) funding councils of the UK funded the JORUM project from October 2002 – July 2005 to establish a repository for learning and teaching materials for use by F/HE institutions in the UK[1]. The JISC are continuing to fund the JORUM to setup a service to the UK FE and HE community and R&D activities will be carried out as part of this work (JORUM Service in Development)[2].

This report forms part of the ongoing Research and Development (R&D) activity of the JORUM project and is a continuing watch on developments in the area of Open Source Software/Free Software (OSS/FS) in the specific context of the JORUM service in development.

It **is** the intention of this report to:

- Review recent developments in the use of OSS/FS that relate to Learning Object Repository (LOR) systems.
- Identify the main issues that need to be considered regarding the feasibility of using OSS/FS for the JORUM repository in the longer term.
- Examine similar projects that have adopted an OSS approach.
- Review existing OSS/FS repository systems.

It **is not** the intention of this report to:

- Propose OSS/FS as an alternative solution for JORUM. Rather, it is the intention to investigate the issues involved in selecting an OSS/FS, compared with a proprietary software solution, which will inform JISC about possible future directions for the JORUM service software solution.

[1] JORUM project website at http://www.jorum.ac.uk/research/archive/index.html
[2] JORUM Service in Development website at http://www.jorum.ac.uk

Conduct a full technical analysis or evaluation of existing OSS/FS repository applications. Access to some repository applications was limited; therefore any attempted evaluation would have been invalid.

2.2 Terminology

The key terms and concepts used in the field of digital repositories frequently have several inconsistent definitions, and this can present difficulties in making satisfactory comparisons between systems.

Although there are numerous possible definitions of the following two terms the JORUM project adopted these definitions[1]:

Learning Object

A learning object (LO) is any resource that can be used to facilitate learning and teaching that has been described using metadata.

Learning Object Repository

A learning object repository (LOR) is a collection of LOs having detailed information (metadata) about them that is accessible via a network or the Internet. In addition to housing LOs, repositories can store 'locations' for objects that are held elsewhere…

Currently, what the report refers to as "generic repositories" such as DSpace, Fedora and Eprints are most often deployed in the following context:

Institutional Repository

The Institutional Repository, as a concept, is to capture and make available as much of the research output of an institution (i.e. a university) as possible. In the first instance this might include material such as research papers and electronic versions of documents such as theses, but may also include many of the digital assets generated by normal campus life, such as administrative documents, course notes, or learning objects.[2]

The following key term has more than one definition in the literature, although there is considerable consensus about the basic meaning given below and used in this report:

Open Source Software/Free Software (OSS/FS)

Open Source Software is software for which the underlying programming code is available to the users so that they may read it, make changes to it, and build new versions of the software incorporating their changes. There are many types of Open Source Software, mainly differing in the licensing term under which (altered) copies of the source code may (or must be) redistributed."[3]

This definition distinguishes OSS/FS from similar terms such as Freeware or Shareware.

2.3 Section Summaries

The report consists of the following sections:

Section 1 - Conclusions: this section presents conclusions drawn from the content covered in the report

Section 2 – Introduction: this section.

[1] JORUM Scoping and Technical Appraisal Study Vol. 1 Overview and Recommendations available from: http://www.jorum.ac.uk/research/publications.html#scope
[2] 'Institutional Repository' article available from: http://en.wikipedia.org/wiki/Institutional_repository
[3] 'Glossary' article of Internet terms used at Wayne State University and available from: www.daahp.wayne.edu/glossary.html

Section 3 – Introductory Survey of Open Source Software: briefly describes some of the main developments in the use of OSS/FS over recent years, focussing on the current wide range of OSS/FS available, especially in the education sector. This review provides a helpful overview of the current status to assist JISC in determining the viability of an OSS/FS solution for the JORUM service in the future. OSS/FS appears to becoming more acceptable model, and various governments are developing policies about this software development model. Considerable organisational support now exists, offering a range of services, such as training and consultation, and the role of the Community is emphasised.

Section 4 - Issues with Using Open Source Software: considers a wide range of questions and issues often raised in conjunction with using the OSS/FS model. Drivers for using this model are examined which highlight the supposed benefits of using OSS/FS rather than proprietary software. UK government policies and those of the JISC are outlined. Most of the issues, such as Total Cost of Ownership (TCO), Reliability/Performance and Standards Implementation are described, giving a balanced view as far as possible.

Section 5 - Digital Repository Technology: examines the possible types of digital repository systems that exist with the requirements of the JORUM service in mind. The various technologies, specifications and toolkits that can underpin a repository system are also explained. Particular reference is made to the importance of using Open Standards and Specifications in the digital repository field.

Section 6 – Overview of International Initiatives and Projects: looks at the existing range of activities internationally. Considerable detail is given about the range of repository systems from an international perspective, showing the contributions of e-Learning programmes and in particular the e-Learning Framework using Web Services to facilitate interoperability.

Section 7 – Review of OSS/FS Learning Object Repositories: examines a number of OSS/FS learning object repositories that provide a closer match for JORUM stated requirements compared to the more generic repositories detailed in section 4.

Section 8 – Selected Case Studies of Open Source Learning Object Repositories: examines a small number of projects that have adopted open source learning object repositories, in much closer detail. These include the well-known Fedora repository system and lesser-known systems such as APOLLO from Canada, the Connexions project in the United States and a UK repository developed in Coventry.

3. Overview of Open Source/Free Software (OSS/FS)

A wide and historical perspective about developments in the use of OSS/FS is helpful in the context of this report in the sense that it shows the degree of maturity and potential of this type of software development model. Looking at the variety of software produced, and the important uses it can be put to, the OSS/FS model has led one prominent author to foresee it becoming mainstream in the near future.[1] Whether or not this does indeed happen, the JORUM project needs to examine recent developments of the OSS/FS model in the light of UK government guidelines regarding giving OS/FS due consideration alongside proprietary software systems.

The key changes in the definition of OSS/FS have mostly related to the type of licence that is attached to this software (see 3.1 and 3.3). The licensing issue is recognised as one of the most important questions to be resolved when embarking on OSS/FS projects and can be crucial to the future development and sustainability of the software project.

The development model referred to as OSS/FS is evolving all the time (see 3.2 History of OSS/FS). The earlier evangelists for Free Software (e.g. Stallman) and those preferring the term Open Source (e.g. Raymond) have had their disagreements in the past but it seems that such disagreements are less prominent today. Perhaps the most significant recent visible developments have been the success of the Linux operating system in capturing a large share of the server market as well as the rapidly increasing popularity of alternative Web browser software such as the Mozilla project's Firefox. There are many other "success" stories in the OSS/FS world. It is a fact that the rise of OSS/FS has coincided with the rapid development of the internet, and in particular the World Wide Web. The use of the Web to co-ordinate the activities of a geographically dispersed group of programmers has been essential to the organisation of substantial OSS/FS projects (e.g. Linux and Apache). Much of the internet now depends on OSS/FS that is known as Middleware[2], especially for communication protocols (e.g. TCP/IP). Such software depends on standards that integrate the diverse and distributed hardware and software that constitutes the modern Internet.

Organisational developments (see 3.4) in the use of OSS/FS seem to be as important as the actual existence of developed software products. There seems to be a trend towards greater sophistication in how projects are organised, supported and maintained these days (e.g. the Apache Software Foundation).

Internationally, there is a growing number of organisations offering various types of support (training, consultation and so on) for OSS/FS users at the organisational and the individual level, some of which are listed in 3.5. Coupled with this increased level of support is the greater prominence of OSS/FS in day-to-day use by a range of well-known companies and organisations. Also, OSS/FS has some big business financial backers including IBM, Sun Microsystems, Novell, Hewlett Packard and other hardware manufacturers. The Linux operating system can run on these types of hardware, and it can be argued that this serves to reduce the overall cost of acquiring such systems. The greater trust shown in OSS/FS by international governments seems to support the view that this model is becoming more mainstream.

The role played by a supportive community of users and developers is examined in 3.6 OSS/FS and the Community. The OSS/FS model relies on self-organised groups of developers compared to the proprietary approach to software development, and it is now recognised that

[1] Wheeler, Brad 'Open Source 2007: How Did This Happen?' article in *EDUCAUSE Review*, vol. 39, no.4 July/August 2004: 12–27 available from: http://www.educause.edu/pub/er/erm04/erm0440.asp

[2] Middleware can be described as software that provides "connectivity" or "glue" functionality, allowing heterogeneous computer systems to communicate. Some examples of Middleware include:

- CORBA (the Common Object Request Brokering Architecture) or Java's RMI (Remote Method Invocation), which allow applications to invoke methods directly upon objects hosted on a remote machine.
- Message-Oriented Middleware such as IBM's MQSeries Messaging or JMS (Java Message Service) implementations, which permit synchronous or asynchronous messaging (using either peer to peer or publish/subscribe communication) between systems.

both users and developers need to collaborate in novel ways to ensure the sustainability of such projects.

In 3.7, consideration is given to the type of organisation that uses OSS/FS with attention given to the education sector in particular. Some international governments have begun to frame policies about using OSS/FS since they perceive advantages in terms of lower total cost of ownership, avoidance of over-dependence on proprietary solutions and, perhaps more importantly, the adoption of Open Standards with this model.

A wide variety of OSS/FS is used in education these days; the HE (Higher Education) sector seems to be the main consumer, although the uptake of Moodle in the FE sector may change this. Education often requires fairly specialist software for supporting teaching and learning; the Virtual Learning Environment (VLE) and Learning Object Repository (LOR) are two relevant examples. There is now a small number of OSS/FS projects focussing on these (and other) areas of interest to educationalists, some of which are detailed in 3.8. The later sections of this report focus specifically on LOR.

3.9 Examines a couple of funding models employed in the education sector, and 3.10 summarises the principal benefits and drawbacks of the OSS approach.

But what is the difference between OSS/FS and the Proprietary software models? To clarify matters:

> Open source refers to a **software licensing model** where the source code of the software is typically made available royalty-free to its users, under terms allowing redistribution, modification and addition, though often with certain restrictions. Open-source programs are often, though not exclusively, developed through a collaborative effort in which a number of persons, often with no formal association with each other, contribute elements of the final software.
>
> Proprietary software refers to the model where the software developed by a commercial entity is typically sold or licensed to a customer in object or executable code, either directly or through channels. The commercial entity often provides support, training, updates and other similar services needed by customers to use that software. The source code of the software may be made available to certain users of the software through special licensing or other agreements, but is usually not distributed to the general public, and may not be copied or modified except in a manner provided for in such agreements.[1]

The models are not mutually exclusive, and companies are increasingly finding ways to embrace both approaches, allowing both to co-exist.

Nevertheless, criticisms are levelled at the Open Source development model and some of these will be discussed in the following section.

3.1 What is OSS/FS?

The meaning of the term OSS/FS has developed since the 1970s reflecting the evolving thinking about these issues: such evolution can be expected to continue.

OSS/FS is also known as Free/Libre Open Source Software (FLOSS) and means that programs developed under such licences:

[1] Hiong, G. S. 'Open source vs. commercial apps: the differences that matter', Oct 2004, available from: http://asia.cnet.com/enterprise/infrastructure/0,39035812,39195508,00.htm

> give users the freedom to run the program for any purpose, study and modify the program and redistribute copies of either the original or modified program (without having to pay royalties to previous developers).[1]

OSS/FS has risen to some prominence in recent months; there has been some increased popular media coverage in connection to the rise of Linux as a feasible alternative to using the all-pervasive Microsoft Windows operating systems on users' desktops.

There is probably greater familiarity with the range of OSS/FS available amongst specialist computing personnel either within academia or industry compared with the typical office worker. The not-for-profit sector of the economy is also taking a greater interest in deploying OSS/FS solutions.[2] This sector believes there are considerable advantages in considering the OSS/FS alternative, especially in the areas of cost-savings and the avoidance of vendor lock-in to proprietary products.

There is an enormous range of OSS/FS available for organisations and individuals to use now without having to depend on proprietary (closed) software suppliers. Software for the organisation usually involves various server–side programs, in particular, Web servers. Software for the individual user includes productive applications for normal office work (e.g. Office Suites) and so on. This situation has changed since the late 1990s when there was a shortage of OSS/FS suitable for the desktop user. The advent of products like Open Office has meant that there is a viable alternative to using the typical MS Office Suite which has become the dominant desktop software for over 90% of the market.

3.1.1 Definitions

Basically, software released under an Open Source licence does not simply mean allowing access to the underlying source code; it must also comply with certain other criteria. The following paraphrases the Open Source Definition (OSD) taken from the Open Source Initiative (OSI) web site[3].

An Open Source licence means:

Free re-distribution of the code, royalty-free
Programs include source-code with distribution as well as compiled code
To support independent peer-review and rapid evolution of the code, the licence must allow modifications of the source code by derived works
To distinguish the original from modified versions of the code, the licence may require that it be distributed as pristine base sources plus patches and given a different name or version number.

The OSI Certified Mark is used to certify that the licence under which the software is distributed conforms to the OSD. The JISC OSS-Watch[4] service has produced an authoritative guide to IPR and licensing issues related to OSS/FS.

Definitions of Free Software (FS) essentially re-phrase the same ideas:

> "Free software" is a matter of liberty, not price. To understand the concept, you should think of "free" as in "free speech", not as in "free beer".[5]

[1] Wheeler, D.A. article 'Why Open Source Software / Free Software (OSS/FS, FOSS, or FLOSS)? Look at the Numbers!' revised as of May 9th 2005 available at :http://www.dwheeler.com/oss_fs_why.html
[2] Murrain et al 'Choosing and Using Open Source Software: A Primer for Nonprofits' available from: http://www.nosi.net/primer/NOSIPrimer.pdf
[3] Open Source Initiative http://www.opensource.org/
[4] 'OSS Watch - Open Source Development - An Introduction to Ownership and Licensing Issues' available from: http://www.oss-watch.ac.uk/resources/iprguide.xml
[5] The Free Software Definition – GNU project available from: http://www.gnu.org/philosophy/free-sw.html

> Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom, for the users of the software:
>
> 1. The freedom to run the program, for any purpose
> 2. The freedom to study how the program works, and adapt it to your needs. Access to the source code is a precondition for this
> 3. The freedom to redistribute copies so you can help your neighbor
> 4. The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. Access to the source code is a precondition for this.
>
> A program is free software if users have all of these freedoms. Thus, you should be free to redistribute copies, either with or without modifications, either gratis or charging a fee for distribution, to anyone anywhere. Being free to do these things means (among other things) that you do not have to ask or pay for permission.[1]

3.1.2 What OSS/FS is not

The term OSS/FS is not to be confused with similar terms such as Freeware, Shareware or Public Domain software. The following definitions are paraphrased from a Wikipedia article[2]:

Shareware usually differs from OSS/FS in that requests of voluntary "shareware fees" are made, often within the program itself, and the source code for shareware programs is generally not available in a form that would allow others to extend the program. Sometimes, paying the fee and obtaining a password results in access to expanded features, documentation, or support. In some cases, unpaid use of the software is limited in time—in which case the software is vernacularly called crippleware.

Freeware is typically free of charge, though regarded as proprietary and has a licence that permits redistribution with some restrictions. It is distributed without source code.

The word "free" in "freeware" refers only to price, whereas “free” in “free software” (FS) refers to freedoms to modify and redistribute the source code that is made available.

Public Domain is one of several variations on the freeware model. Software in the public domain has no copyright and therefore may be distributed without charge. Freeware is usually copyrighted and its licence may restrict certain activities.

3.2 History of OSS/FS

Although it may seem that developing and using Open Source Software or Free Software (OSS/FS) is a new trend the fact is that, from the very earliest days of computing (approx 1945-1975), programmers actually did share their code with each other. The Unix operating system was developed by AT&T researchers and was distributed as source code, with modification rights, for a nominal fee.

> However, as years progressed, and especially in the 1970s and 1980s, software developers increasingly closed off their software source code from users. This included the Unix system itself …[3]

In reaction to these developments Richard Stallman established the Free Software Foundation (FSF). Sponsored and organised by the FSF, the GNU project developed a Unix-like operating system, which, with the addition of the Linux kernel code written by Linus Torvalds, became the

[1] ‘The Free Software Definition’ article available from: http://www.gnu.org/philosophy/free-sw.html
[2] ‘Freeware’ article available from: http://en.wikipedia.org/wiki/Freeware
[3] Wheeler, D ‘Why Open Source Software / Free Software (OSS/FS, FOSS, or FLOSS)? Look at the Numbers!’ revised May 9th 2005’ available at http://www.dwheeler.com/oss_fs_why.html

now-familiar Linux operating system. A major legal innovation by Stallman was the GNU General Public Licence (GPL)[1], a widely popular OS software licence that was adopted by Torvalds. Terminology has changed, and the term "Free Software" gave way to "Open Source".

3.3 Licensing

There are two principal OSS/FS approaches to licensing - the GNU General Public Licence (GPL) and the Berkeley Software Distribution (BSD) Licence[2]:

Under the GPL, all derivative works of the software must be licensed and distributed on the same terms as the original software. Source code subject to the GPL cannot be disassociated with that license and permanently remains as such. Under the BSD License, developers are allowed to integrate the licensed software with the developers' own source code to create new products with few restrictions.

The BSD License, for example, allows programmers to use, modify and redistribute source code and binaries of the original software. But unlike the GPL approach, programs containing code subject to the BSD License do not have to be distributed under the BSD License. Derivative works can be distributed either in an open source manner, or in a more traditional "commercial" license.

Another characteristic of the GPL approach relates to distribution. The GPL prohibits charging money for the distribution of source code, other than to cover the administrative cost of copying and shipping. However, charging fees for system setup, system management, support, maintenance and other related services is permitted.[3]

For OSS/FS the question of licences is very important and four major licences are used:

GNU General Public License (GPL)
GNU Lesser (or Library) General Public License (LGPL)
MIT (aka X11) license
BSD-new license

The Open Source Initiative[4] - an organisation that manages and promotes the Open Source Definition for the good of the community - refers to these four licences as the classic open source licences. According to the Open Source Initiative website[5] there are almost 50 separate licences from which to choose and projects are warned not to create their own licences unnecessarily.

All of the 'classic' licences (the GPL, LGPL, MIT, and BSD-new) **permit the commercial sale and the commercial use of the software**, and many such programs as sold and used that way. Of these four licence types the General Public Licence (GPL) is by far the most popular licence. David Wheeler argues strongly that GPL-compatible licences should be used (like the 'classic' licences above) because there is a significant risk that, without such a licence, projects will not receive enough support from other developers to sustain it, since many OSS/FS developers prefer the GPL[6].

This has been a hotly-debated topic in recent years, one of the main debates have been between the BSD (old version) and GPL approaches to licensing.

---

[1] GNU General Public Licence http://www.gnu.org/licenses/gpl.html
[2] The BSD licence at http://www.opensource.org/licenses/bsd-license.php
[3] Hiong, S.W. 'Open source vs. commercial apps: the differences that matter' article October 2004, available from: http://asia.cnet.com/enterprise/infrastructure/0,39035812,39195508,00.htm
[4] The Open Source Initiative http://www.opensource.org/
[5] The Approved Licenses http://www.opensource.org/licenses/
[6] Wheeler, D 'Make Your Open Source Software GPL-Compatible. Or Else.' revised February 16, 2005 article available from: http://www.dwheeler.com/essays/gpl-compatible.html

3.4 Organisational Developments in the OSS/FS Model

The organisational and managerial models to support OSS/FS have developed in terms of increased sophistication since the earliest days. The informal model employed by early Linux developers has matured into a more conventional model as exemplified by the Apache Software Foundation and companies such as Orixo which aim to offer consultancy and advisory services about using OSS/FS to potential 'customers'.

OSS consultancy services are becoming more popular where suppliers can de-couple code from service. Instead of charging for the licence to use the software (not to own it) companies offering OSS/FS consultancy and training services see a viable commercial business model for themselves. Users find that they often have to pay for training and certain forms of software support, whichever software development model is used.

The original enthusiasts for OSS/FS were often motivated by various ethical and political ideals concerning notions of openness of source code and the perceived limitations of commercially developed code. OSS/FS has relied, to a remarkable extent, on the development of a community of like-minded enthusiastic programmers and testers to produce products such as Linux, and the Apache web server.

Organisation and management approaches are important for proprietary as well as OSS/FS but, arguably, they are more important for OSS/FS. In an influential essay[1] Raymond, E.S. (1997) popularised Torvald's approach with a large number of small, incremental releases, and a large number of developers sending in patches for proposed improvements. The releases need to compile and run (to some extent), so that developers can test and improve them. This is a highly de-centralised model, taking advantage of the Internet for communication.

The Apache web server project, in contrast, was led by a co-located group.

> Apache originated in early 1995 as a series of enhancements to the then-popular public domain HTTP daemon developed by Rob McCool at the National Center for Supercomputing Applications, or NCSA. Rob McCool had left NCSA in mid 1994, and many Webmasters had become frustrated with a lack of further development. Some proceeded to develop their own fixes and improvements. A small group coordinated these changes in the form of patches and made the first official release of the Apache server in April 1995, hence the name A PAtCHy server (Laurie, 1999).
>
> The Apache Group is currently a core group of about 20 project contributors, who now focus more on business issues and security problems. The larger user community manages mainstream development. Apache operates as a meritocracy, in a format similar to most open-source projects. Responsibility is based on contribution, or "the more work you have done, the more work you are allowed to do." (The Apache Group, 1999) Development is coordinated through the *new-httpd* mailing list, and a voting process exists for conflict resolution.[2]

The Apache Group, having grown considerably since its inception, re-named itself the Apache Software Foundation in 1999.

3.5 OSS/FS Organisations

There are numerous groups that provide important services to the community of OSS/FS projects currently in existence. Some of these are listed here:

[1] Raymond, E.S. 'The Cathedral and the Bazaar' article available from: http://www.catb.org/~esr/writings/cathedral-bazaar/
[2] Johnson, K "Open-Source Software Development" article available from: http://chinese-school.netfirms.com/index.html

3.5.1 OSS-Watch

This is the JISC Open Source Software Advisory Service[1] set up to provide unbiased advice and guidance about free and open source software for UK further and higher education. This is a very usable source of information for strategic IT decision-makers, IT managers and technicians.

> OSS Watch develops best-practice guidelines, investigative reports, and briefing materials. As well as a web-based clearing-house for up to date information, OSS Watch organizes national conferences, expert workshops and offers focused assistance for institutions and software projects.[2]

3.5.2 The Apache Software Foundation (ASF)

This organisation supports the development of a number of OSS/FS projects developed by the Apache community.

> The Apache Software Foundation provides organizational, legal, and financial support for a broad range of open source software projects. The Foundation provides an established framework for intellectual property and financial contributions that simultaneously limits contributors potential legal exposure.
>
> …the Foundation has been incorporated as a membership-based, not-for-profit corporation in order to ensure that the Apache projects continue to exist beyond the participation of individual volunteers.[3]

3.5.3 SourceForge

> SourceForge's mission is to enrich the Open Source community by providing a centralised place for Open Source developers to control and manage Open Source software development.[4]

3.5.4 CampusSource

CampusSource[5] is funded by the Ministry of Science and Research of the Federal State of North Rhine-Westphalia (Germany) whose aim is to set up cooperative processes for the development of software systems and modules as well as the creation and operation of an infrastructure for computer and network based teaching and learning at universities. The efforts of single university projects will be brought together and the Open Source platforms as technical requirement of a virtual university will be provided for use and further development to everyone interested.

CampusSource is in many ways similar to the more familiar Open Knowledge Initiative developing at MIT[6] that

> develops specifications that describe how the components of an educational software environment communicate with each other …
>
> [and the specifications] address broad interoperability agreements that allow for adaptations and further specification by communities of practice.

[1] OSS Watch http://www.oss-watch.ac.uk/
[2] OSS Watch http://www.oss-watch.ac.uk/
[3] http://www.apache.org/foundation/
[4] http://sourceforge.net/index.php
[5] http://www.campussource.de/org/
[6] Open Knowledge Initiative http://www.okiproject.org/

3.5.5 OpenAdvantage

OpenAdvantage[1] is a Centre of Excellence in the West Midlands of England, is one of a growing number of organisations specialising in OSS/FS who provide a free consultancy service. This Web site uses the OSS/FS Plone Content Management System.

> Open Source software can be hard to evaluate - you don't get brochures, there aren't salesmen to give you demos, and there are no glossy adverts to impress. However most open-source projects do have a common structure, which, once you understand it, can be used to help make evaluations.[2]

3.5.6 Open Source Consortium

The Open Source Consortium has been formed to represent the Open Source business community in the UK. They are

> a not-for-profit organisation which guarantees the quality of open source deployments in the public sector by setting professional standards and bonding its members.[3]

Their purpose is to do business with the UK government and public sector and to represent the views of the emerging industry using OSS/FS.

3.5.7 Open Source Software Institute

The Open Source Software Institute is US-based and focuses on:

> corporate, government and academic representatives whose mission is to promote the development and implementation of open-source software solutions within U.S. federal, state and municipal government agencies and academic entities[4]

3.5.8 Schoolforge-UK

This is the UK branch of SCHOOLFORGE[5] a US organisation, which appears to have many affiliated members. Schoolforge-UK is itself only a small voluntary organisation that arose from the Open Source in Education Conference (2003) hosted by Anglia Polytechnic University.

> Schoolforge-UK's mission is to bring together individuals and organisations that advocate, use, and develop open resources for UK schools and colleges.[6]

They advocate the use of Free/Open Source Software (FOSS) in education regarding this as the norm rather than the exception.

3.6 OSS/FS and the Community

Previous sub-sections have raised the issue of organisational developments in OSS/FS projects and activities. If it is important for the success of OSS/FS projects that active,

---

[1] http://www.openadvantage.org/
[2] 'Evaluating open source software' article available from: http://www.openadvantage.org/articles/oadocument.2005-03-10.5941460159
[3] Open Source Consortium http://www.opensourceconsortium.org/index.html
[4] Open Source Software Institute http://www.oss-institute.org/
[5] SCHOOLFORGE http://www.schoolforge.net/
[6] Schoolforge-UK http://www.schoolforge.org.uk/index.php/Schoolforge-UK:About

supportive communities of users and developers exist then it is also important that we know more about how OSS/FS Communities are built and sustained. This section draws on the July 2005 OSS Watch Conference which examined some of these important questions including:

How do open source software projects organise themselves?
Why do some projects grow rapidly?
How does the licence choice affect the kind of community that develops?
What is the relationship between developers and end-users?
How important is leadership?[1]

Sebastian Rahtz of OSS Watch pointed out at this conference that OSS/FS projects can develop in various ways; an unfortunate reality is that the project can simply die out. Many OSS/FS projects can reside in the SourceForge repository, more or less ignored by the community, receiving no further development. The originators of some OSS/FS may simply move on to other interests, change jobs and so on, leaving the 'community' to continue developing the project if so desired. Sustaining development of certain OSS/FS projects can be a problem. One way to sustain fledgling OSS/FS projects is develop a real community of users and developers who continue to work on the project because of its relevance to them. Two examples follow:

3.6.1 Examples of OSS/FS Communities

- **Moodle.** The Moodle project has certainly benefited from the presence of a supportive community. Originally built by a single programmer (Martin Dougiamas, Australia) this impressive VLE software has attracted the active help of many teachers who make a range of contributions (e.g. suggestions for improvements, documentation, software help). The OSS/FS licence that allows users to freely download the software has led to there being over 4000 registered Moodle sites around the world. Interested users can inspect the source code and become involved in the improvement of the functionality of the product – some by contributing their own code for inclusion in the product. To encourage the virtual communities that have grown to support Moodle, more traditional methods of community-building have now begun to occur (e.g. the Moodlemoot, Oxford 2005, organised by Sean Keogh). Moodlemoots allow Moodle users to meet face-to-face in order to talk and share ideas.

- **Apache.** Andrew Savory discussed the community aspect of the Apache Software Foundation (ASF) in his talk 'Life and times in the Apache community'. Different OSS/FS communities use different strategies: the ASF describe themselves as a Meritocracy, who can have the role of Users, Committers and Members.

3.6.2 Users as Developers?

In OSS/FS projects there is often much less of a distinction between users and software developers, although it would be easy to exaggerate this point: clearly, for most projects, users will have neither the skill nor the inclination to contribute source code improvements. Nevertheless, the ability for anyone to inspect the source code of projects has allowed users to feel they can participate in the software development process by contributing their suggestions in forums and on mail lists. According to Sean Keogh speaking at the same conference, the author of Moodle (Martin Dougiemas) has been known to respond to requests for increased functionality of Moodle within 24 hours. When connectivity to an email server was requested Dougiemas emailed the new code to the user for him to test.

The ASF arranged events whereby users can meet together to solve software bugs posed as a challenge to them by the organisers. Very often, users of Apache will be programmers who

[1] 'Building Open Source Communities' OSS Watch Conference July 2005 available from: http://www.oss-watch.ac.uk/events/2005-07-04/

work for other software companies who need improvements to the Apache software. Only Committers are able to modify the source code held by the ASF.

3.6.3 Communities of Practice

One OSS Watch presentation compared the participation in the OSS/FS Bodington VLE as being similar to a Community of Practice[1] (CoP) defined as:

- informal networks that emerge from a desire to work more effectively or to understand work more deeply among members of a particular speciality or work group.
- small groups of people who've worked together over a period of time and through extensive communication have developed a common sense of purpose and a desire to share work-related knowledge and experience.
- communities of apprentices where newcomers learn by gradually going from peripheral participation to full participation in the community.

[where] learning is presented as a process of social participation in a community.[2]

The Moodle community has attracted many teachers from the primary and secondary sectors who have found the software to be useful in their teaching. The software seems to address some of their practical problems and as such they feel able to participate in its continuing development. Similar remarks can be made about the Apache Software Foundation. Apache Cocoon attracts Web programmers who need the functionality offered by the various projects supported by Cocoon and the ASF. People who become involved in OSS/FS projects report various motivations for participation. For the software developers, the desire to learn technical skills is usually high on their personal list.

There is now some published research on how these communities develop and function drawing upon case studies of some of the more prominent OSS/FS projects over the years.[3] The theme of Community and Culture was addressed by Helen Sharp of the Open University in a presentation entitled 'Co-located agile development'. Her review of empirical studies of software development using the Extreme Programming methodology showed that a prime value of the programmers involved providing value to customers using techniques of "Customer Collaboration". In certain respects, this approach may be compared with the collaboration taking place between Users and Developers of OSS/FS.

The software industry is very young and in many ways not like typical industries. One noticeable difference is the heavy dependence upon a professional workforce of young people trained in esoteric skills. In reality, there will be a great deal of variation in how software companies operate, depending on factors such as the size of the company, its annual turnover and the guiding philosophies of its original founders. Even the proprietary model can be said to be supported by a "community" – the body of people working under the umbrella of that organisation. However, the OSS/FS model seems to rely on the voluntary contributions of a distributed workforce in a more obvious manner than does the proprietary model. This workforce forms the OSS/FS Community.

3.6.4 Java Communities

There are many communities in existence now, serving the needs of sub-groups of Java users. Although the Java language is not itself OSS/FS, (though the Apache Foundation are now planning to develop such a version) Sun Microsystems actively support community involvement

[1] Smith, M. 'communities of practice' encyclopaedia article available from: http://www.infed.org/biblio/communities_of_practice.htm
[2] OSS Watch 'Building Open Source Communities' 2004 available from: http://www.oss-watch.ac.uk/talks/2004-11-19-bodington/index.xml.ID=body.1_div.9
[3] Kim, E.E. 'An Introduction to Open source Communities' April 2003 available from: http://www

through their website[1] as well as java.net, the new central meeting place for developers and Java technology enthusiasts and existing communities across industries, platforms, and interest groups.

Sun have set up the Global Education and Learning Community[2] ((previously the Java Education Learning Community) to be:

> a place to find, develop and share Java-related open source educational tools, open learning standards implementations, and open source course learning materials.[3]

The GELC community consists of a wide range of people including teachers, programmers, students and researchers.

Sun's vision is that members of the GELC will share resources, leverage best practices, and work on global collaborative education and e-Learning projects. There is a focus on open standards and specifications (see section 4.4.3) led by the IMS, SIF, SCORM and METS.

GELC projects include:

- Open desktop tools
- e-Learning framework
- Academic research tools
- Learning resources

JA-SIG (Java in Administration Special Interest Group)[4] is an independent organization promoting the use of Java technologies and architectures within the higher education community. JA-SIG supports the development and adoption of low cost, flexible, open source solutions that adhere to best practices and open standards. The purpose of JA-SIG is to share experiences of building applications with Java and to support a common architecture upon which shareable, open source components can be built.

The JA-SIG organises conferences, acts as a place on the Web to facilitate the sharing of Java components and is active in the development of a free, open source, open standard portal for higher education, uPortal (see 2.8.5 for further details).

There is considerable support for Java language developers of e-Learning tools and architectures using an OSS/FS development model.

3.7 Who Uses OSS/FS?

In the early days, using OSS/FS was restricted to enthusiasts; now, international governments show increasing enthusiasm for the idea of using OSS/FS in even mission-critical situations. From official surveys of the uptake of Open Source (the FLOSS survey[5]) there are some emergent trends. Certain governments are more enthusiastic about FLOSS than others; within the European Union (EU) Holland are willing to trust OSS/FS to manage some of their flood control systems, as well as waste management processes. The Spanish region of Extramadura has invested heavily in Linux boxes and has put OSS/FS in many official desktops. Outside Europe, the US Department of Defence relies on a specially secure version of Linux and China and Brazil are keen users of OSS/FS.

[1] http://java.sun.com/
[2] GELC http://community.java.net/edu-gelc/
[3] http://www.oss-watch.ac.uk/events/2004-06-03/heath.pdf
[4] http://www.ja-sig.org/
[5] FLOSS survey available from: http://flosspols.org

Governments are framing policies to ensure that OSS/FS is considered whenever new software procurement is planned.

As part of government, the education sector has shown perhaps the most interest in using OSS/FS.

3.7.1 OSS/FS in the Education Sector

In the UK, and internationally, interest in using OSS/FS in education is hardly new although the skills-base is generally higher in the HE sectors compared with the FE and pre-16 sectors.

Internationally, then, there are signs of the increased importance being given to exploring the OSS/FS alternative; for example, the Spanish region of Extramadura has installed entirely OSS/FS for their servers and desktops in their school system. By building on the Linux operating system and installing desktop software such as OpenOffice this became a highly cost-saving process for the Spanish. The use of OSS/FS by various governments around the world (especially in some of the 'developing' nations), is probably motivated by the apparent savings from the cost of licensing such software, as much as for the ability to modify the underlying, and available, source code.

For a recent review of why educational institutions are considering choosing OSS/FS learning environments the reader is referred to the Christina Smart article for JISC e-Learning Focus[1]. This article refers to a recent newsworthy report[2] from BECTA "Open Source Software in Schools: A study of the spectrum of use and related ICT infrastructure costs" indicating that, although the implementation of OSS/FS in schools needs careful planning and support, the use of OSS/FS can offer a cost-effective alternative to proprietary software.

For the post-16 sector there are OSS/FS versions of Virtual Learning Environments (VLE) that are increasingly being used. These include Moodle and Bodington. There are many more proprietary VLE products on the market that the HE sector can select but the cost of funding annual licences is a cause for concern.

As well as software specifically designed to support online learning (such as a VLE, or LCMS) there are also numerous proprietary Content Management Systems (CMS) in use by the Business and Education sectors. Proprietary CMS can be very expensive to pay for over a period of time.

3.8 Some OSS/FS products used in Education

There are a number of OSS/FS products that currently fit into the e-Learning landscape. These range from the more complex, institutional Virtual Learning Environment (VLE) systems like Bodington and Moodle to the smaller, specialised tools such as RELOAD.

3.8.1 Moodle

Moodle[3] is a software package designed to help educators create quality online courses. One of the main advantages of Moodle over other systems is a strong grounding in social constructionist pedagogy.

The current version of Moodle, 1.4.4, was released on 10th March, 2005. As of March 2005, more than 3100 sites from more than 115 countries have registered their Moodle installation. Moodle's approach is said to be related to the Learning Object's approach.[4]

[1] Smart, C. 'Choosing Open Source Solutions' May 2005 article available from: http://www.elearning.ac.uk/features/oss#ref1
[2] BECTA report available from: http://www.becta.org.uk/corporate/publications/documents/BEC5606_Full_report18.pdf
[3] Website at http://moodle org
[4] Taken from Wikipedia at: http://en.wikipedia.org/wiki/Moodle

Moodle has been adopted by The Open Polytechnic and a consortium of six polytechnics in New Zealand. The 'Polytechs migrate to Moodle' article[1] provides the following information:

> The Open Polytechnic is leading a consortium of polytechs in a $1 million project to deploy open-source e-learning software. The Open Source Virtual Learning Environment project, funded by the Tertiary Education Committee, is believed to be the largest deployment of "Moodle" software to date globally.
>
> Project manager Richard Wyles says the polytechs started investigating open source as a way of cutting costs. He estimates that using Moodle instead of proprietary software will save the Open Polytechnic about $50,000 every year in licence fees alone.
>
> "Moodle's got everything proprietary systems have got and more," Mr Wyles says. It is designed to support traditional in-class learning as well as correspondence courses.
>
> Mr Wyles says Moodle is also more flexible than competing proprietary programs such as WebCT or Blackboard, since modules can be attached to Moodle as they're developed.[2]

3.8.2 Bodington

Bodington[3] is a free virtual learning environment supporting teaching and learning across the entire range of learning institutions in the UK and worldwide, e.g. the Universities of Leeds, Oxford, Manchester; UHI Millennium Institute, Eton College, Yorkshire Coast College and St Andrews College Singapore.

The Bodington course management system was developed out of the early work of Jon Maber at the University of Leeds. The system is Java-based and runs on the latest version of Tomcat (version 5.5). The previous version, Bodington 2, had been OSS/FS for some time as is the latest version, Bodington III, which is one of the e-Tools[4] funded by the JISC distributed e-learning strand. The project aims to develop a framework that implements extended Java Web Application Specification containers, incorporating file system, authentication and collaborative work facilities. Important e-Learning tools (e.g. the open source TOIA assessment engine), addressed in the JISC e-learning framework, will be integrated into Bodington III as well as the OSS/FS LAMS software (see below) and e-portfolio functionalities.

3.8.3 Sakai

The Sakai project[5] also OSS/FS, concerns the building of a Virtual Research Environment (VRE) and is a Course Management System (CMS) whose latest release is version 1.5 as of March 5th 2005.

> The Sakai Project's primary goal is to deliver the Sakai application framework and associated CMS tools and components that are designed to work together. These components are for course management, and, as an augmentation of the original CMS model, they also support research collaboration. The software is being designed to be competitive with the best CMSs available.
>
> The tools are being built by designers, software architects and developers at different institutions, using an experimental variation of an open source development model called the community source model. To provide a support system for institutions that want to be involved in the Sakai Project, either by adopting Sakai tools or by developing tools for inter-institutional portability, the Sakai Project has also formed the Sakai Educational Partners Program (SEPP) and the Sakai Commercial Affiliates Program.

[1] 'Polytechs migrate to Moodle' article available from: http://www.stuff.co.nz/stuff/0,2106,3251313a28,00.html
[2] 'Polytechs migrate to Moodle' article available from: http://www.stuff.co.nz/stuff/0,2106,3251313a28,00.html
[3] Bodington http://bodington.org/index.jsp
[4] Bodington III Project http://www.elearning.ac.uk/del/etools/bodington
[5] Sakai http://www.sakaiproject.org/cms/

> The Sakai Project has its origins at the University of Michigan and Indiana University, where both universities independently began open source efforts to replicate and enhance the functionality of their existing CMSs. Soon after, MIT and Stanford joined in and, along with the Open Knowledge Initiative (OKI) and the uPortal consortium, and a generous grant from the Mellon Foundation, they formed the Sakai Project.[1]

3.8.4 Sofia

The Sofia project[2] is a repository system that was inspired by Sakai. It includes the provision of 'open content' (following the MIT Open Courseware Initiative). Sofia open content will be published in ETUDES-NG[3], a CMS that will be based on the Sakai open source software. Sofia encourages the publication and free exchange of community college-level course materials on the Web.

3.8.5 uPortal

The uPortal website[4] states that:

> uPortal is an open-standard effort using Java, XML, JSP and J2EE. It is a collaborative development project with the effort shared among several of the JA-SIG member institutions.

The uPortal group

> sees an institutional portal as an abridged and customized version of the institutional Web presence... a "pocket-sized" version of the campus Web. Portal technology adds "customization" and "community" to the campus Web presence. Customization allows each user to define a unique and personal view of the campus Web. Community tools, such as chat, forums, survey, and so on, build relationships among campus constituencies.[5]

Interest in the use of portal technology has definitely increased in recent years. Current production users of the OSS uPortal include many Universities in North America, where the software was initially developed, as well as several UK Universities (e.g. Hull, Bristol, Edinburgh, and Nottingham).

3.8.6 LAMS

LAMS, the IMS Learning Design 'inspired' Learning Activity Management System from Australia is now OSS/FS. It provides teachers with a highly intuitive visual authoring environment for creating sequences of learning activities. These activities can include a range of individual tasks, small group work and whole class activities based on both content and collaboration.[6] Launched on April 13th, 2005, LAMS is freely available under the GNU GPL.

LAMS integrates with other e-learning systems such as Learning Management Systems, Learning Object Repositories, Portals and so on. Learning Design focuses more on the educational processes that occur during learning, rather than on the educational content being acquired by the learner. The IMS Learning Design specification draws attention to the educational activities which learners engage in during the process of learning described using the Educational Markup Language (EML) developed in the Netherlands. XML is used to describe how people in roles (teachers, students) carry out the learning activities using particular learning resources. There is now an OSS/FS tool – CopperCore – which serves as an IMS Learning Design Engine to create Learning Designs (LD). Other OSS/FS tools for creating LD include RELOAD which was recently extended to support this specification.

---

[1] http://www.sakaiproject.org/index.php?option=com_content&task=view&id=103&Itemid=208
[2] Sofia http://sofia.fhda.edu/index.htm
[3] ETUDES-NG http://foothillglobalaccess.org/etudes2
[4] uPortal at http://www.uportal.org/
[5] uPortal at http://www.uportal.org/
[6] LAMS at http://www.lamsinternational.com/

3.8.7 RELOAD

RELOAD (Reusable Learning Object Authoring and Delivery) [1] is a JISC funded project (initially funded under JISC's X4L Programme, which also funded the JORUM project) that develops tools to facilitate the use of emerging Learning Technology Interoperability specifications such as those produced by ADL and IMS. Managed at the University of Bolton, the primary aims of this project are to:

> facilitate the creation, sharing and reuse of learning objects and services
> enhance the range of pedagogical approaches realisable through the use of lesson plans

The first strand of the project focuses on developing a set of OSS/FS tools: a Content Package Editor, Runtime Delivery System, and the Colloquia VLE. The Content Package Editor supports all aspects of the latest and previous versions of IMS Content Packaging, including IMS Meta-data with IEEE LOM (Learning Object Metadata) vocabulary. It enables content to be aggregated into different structures and tagged with meta-data for exchange between systems and delivery to learners. RELOAD has continued to develop and is now at version 2.1.0; these are further refinements to its ability to handle the IMS Learning Design specification, which has now become the main focus of their development work.

3.8.8 TOIA[2]

TOIA (Technologies for Online Interoperable Assessment) was also initially funded under JISC's X4L Programme, which also funded the JORUM project.

The principal aims are to develop an assessment authoring system that fully implements the IMS Question and Test Interoperability Specifications, whilst ensuring that a broad range of question authoring, test construction and administration requirements are met. TOIA is developing software to interact with its QTI databases for questions, tests and results that could lead the way in defining a scalable infrastructure for fully interoperable test banks in the UK.

TOIA is now part of the JISC Distributed e-Learning programme known as TIP (Tools Integration Project) that aims to integrate two other OSS/FS projects: the Bodington VLE and the LAMS learning design environment under a single sign-on. The project will also develop web services that allow the three software systems to provide and consume each other's services. Integration of these tools was demonstrated in July 2005.

The New Zealand Ministry of Education has recently announced the integration of LAMS and Moodle[3]. The Moodle/LAMS integration will also achieve single sign-on between the systems, so that only a single username and password is needed to give each user access to the two systems.

3.8.9 Open Source Portfolio Initiative (OSPI)

There is growing international interest in the concept of the ePortfolio for students.

> An eportfolio can be a web-based information management system that uses electronic media and services. The learner builds and maintains a digital repository of artifacts, which they can use to demonstrate competence and reflect on their learning. Having access to their records, digital repository, feedback and reflection students can achieve a greater understanding of their individual growth, career planning and CV building.[4]

The IMS view is that

> ePortfolios are collections of personal information about a learner that represent accomplishments, goals, experiences, and other personalized records that a

[1] RELOAD website http://www.reload.ac.uk/
[2] TOIA website http://www.toia.ac.uk/
[3] LAMS/Moodle integration http://www.lamsfoundation.org/integration/moodle/
[4] 'ePortfolios Portal: What is an ePortfolio?' article available from: http://www.deskootenays.ca/wilton/eportfolios/whatitis.php

learner can present to schools, employers, or other entities. Typical uses of ePortfolios go beyond the traditional concept of a transcript to include applying for jobs, designing personalized learning, and tracking career planning.[1]

The IMS ePortfolio Specification[2] functions to enable the interoperability (import and export) of parts of the portfolio between portfolio systems and organisations.

The Open Source Portfolio Initiative (OSPI)[3] is a community of individuals and organisations collaborating on the development of the leading non-proprietary, OSS/FS electronic portfolio software available. The recent release of the Open Source Portfolio Version 2 is a major step forward for the use of educational portfolios.

3.8.10 Plone

A very popular OSS/FS version of a CMS is Plone[4], used by a number of Universities throughout the world. The Plone project was started in 1999 by Alan Runyan, Alexander Limi, and Vidar Andersen. It has quickly grown into one of the most popular and powerful open source content management systems in the world. In 2004, the Plone Foundation was formed to protect and promote the use of Plone. Plone is released under the GNU[5] General Public Licence[6] and is built on top of the open source application server Zope and the accompanying Content Management Framework, which has thousands of developers around the world supporting it. A recent educational application of Plone is eduPlone (see section 6.5 for further details).

3.8.11 Wikis

A Wiki is a piece of server software that allows users to freely create and edit Web page content using any Web browser. Wiki supports hyperlinks and has a simple text syntax for creating new pages and crosslinks between internal pages on the fly. Many Wikis use OSS/FS such as Perl, python, Zope and so on, though almost any language can be used to create a Wiki.

Wikis are often used in education to support a wide variety of collaborative projects, in an impressively large number of languages. The term *wiki* (derived from the Hawaiian word for "quick") is applied to a diverse set of systems, features, approaches, and projects. One of the best-known wiki-projects on the Web is the Wikipedia[7] with 656,525 articles in the English language version (on 28.07.05).

For some details of how Wikis are actually being used in Education, the University of British Columbia's pilot project has included academics linking a wiki to a WebCT course to support the design process of the online course with colleagues. Activities included: building reference lists, course outlines, brainstorm instructional strategies, and complete systematic questionnaires.[8]

The Wikimedia Foundation Inc. is the parent organisation of various free-content projects that are useful in the education sector. The goals of the foundation are to maintain and develop free-content, wiki-based projects and to provide the full contents of those projects to the public free of charge. In addition to managing the already developed multilingual general encyclopaedia Wikipedia, there is a multi-language dictionary and thesaurus named Wiktionary, an encyclopaedia of quotations named Wikiquote and a collection of e-book resources aimed specifically toward students (such as textbooks and annotated public domain books) named Wikibooks.

---

[1] IMS Global Learning Consortium: ePortfolio Specification at http://support.imsglobal.org/eportfolio/
[2] IMS ePortfolio Specification http://www.imsglobal.org/ep/
[3] The Open Source Portfolio Initiative at http://www.theospi.org/
[4] Plone http://plone.org/
[5] GNU http://en.wikipedia.org/wiki/GNU
[6] GPL http://en.wikipedia.org/wiki/General_Public_License
[7] Taken from Wikipedia at: http://wikipedia.org/
[8] 'Taking a Walk on the Wiki Side : Campus Technology' article, April 2004 which was originally published in Syllabus and is available from: http://www.campus-technology.com/article.asp?id=9200

3.8.12 Summary

This section has demonstrated that there is currently a wide variety of OSS/FS products being used in the higher education, and to some extent further education, sector. Some types of products have attracted more interest and development in under the OSS/FS model than others. A possible explanation might be that well-established types of products such as VLEs and CMSs, with proven markets, have attracted the attentions of enthusiastic software development teams. In contrast, less established products, like Learning Object Repositories, are currently fewer in number in terms of the development of OSS/FS systems. It may be that all of these products are capable of attracting a large-enough user-base so that it becomes worthwhile to develop OSS/FS versions of the software. However, as LOR are niche-products (at least currently) this may always limit the number of OSS/FS versions that are created.

3.9 Some OSS Funding Models Employed in the Education Sector

3.9.1 Community Source

Sakai, uPortal and OSPI use the term Community Source to refer to the model of investment and governance used by their projects. Community Source is:

> community investments for community outcomes in the open source tradition[1].

Many of the investments in community source represent institutional investments from the Higher Education sector. The Sakai Project, the Open Source Portfolio, and uPortal were all seeded with a generous gift from the Andrew W. Mellon Foundation.

3.9.2 Open Access/Open Archives

Through the Open Society Initiative, supported by the philanthropist George Soros, funding is given to many initiatives to make scholarly journals freely available to the international community. In particular, the Budapest Open Access Initiative.[2]

Many institutions are realising that there is no need to spend large amounts of (public) money to pay for the published research papers of their academics to be hosted by commercial journals. Online publications committed to Open Access allow institutions to host their own publications and obtain shared access with other institutions across the world.

3.10 Benefits and Drawbacks of using OSS

Apart from reviewing some of the OSS/FS in the news, such as the increasing popularity of Moodle VLE, the Christina Smart article for JISC e-Learning Focus[3] includes an interview with Stuart Yeates from the JISC OSS Watch about the benefits and drawbacks of using OSS/FS.

Importantly, the article notes that a formal software procurement process should be undertaken by the institution so that whether OSS/FS or proprietary, the software is chosen for the right reasons.

The main benefits and drawbacks of using OSS/FS as discussed in Smart's article for the JISC e-Learning focus, are summarised below:

3.10.1 Benefits

- the software developed uses open standards which is good for interoperability
- the ability to tailor the system completely to local needs
- proprietary software typically charges in steps for licensed use - "For example, the first 500 users might cost x thousand pounds per year, and every hundred users above that adds a

[1] Brad Wheeler cited at the Sakai project website: First Community Source Week, 8th – 14th June, 2005 in Baltimore http://www.sakaiproject.org/index.php?option=com_content&task=view&id=250&Itemid=222
[2] Open Access Initiative http://www.soros.org/openaccess/read.shtml
[3] JISC e-Learning Focus, Choosing Open Source Solutions, Christina Smart (July 2005) http://www.elearning.ac.uk/features/oss

further increment. It doesn't matter whether those users use the software 8 hours a day 46 weeks a year or one hour every second week during term."

- lack of surprises – "Open source licences are free and perpetual, so a licence fee increase cannot happen. There is no vendor to go bankrupt, get taken over or discontinue the system and leave institutions in the lurch. These factors considerably decrease the chances of unwelcome surprises down the track. It is, however, true that an institution which has never really used open source in the past will find their first use of it a learning experience."
- cooperative development – "Open source removes the commercial imperative to compete, enabling genuine cooperation between developers and institutions, among developers and between projects. For example, Bodington developers are attending the upcoming MoodleMoot in Oxford to demonstrate some of the work they're doing on getting Bodington and Moodle to work together. In particular, I believe, they're using Bodington's very flexible access permissions system and making it accessible within Moodle, which uses course-based permissions." [1]

3.10.2 Drawbacks:

- The main drawback is that OSS/FS is usually free at the point of acquisition – "confusing for institutions accustomed to paying an annual licence fee for software". This is the main difference between OSS/FS and proprietary systems.
- Sustainability of open source products – a genuine anxiety of institutions "that software developed by the "community" might simply vanish". However, "if a significant number of institutions deploy a system such as the Bodington VLE, and even a tiny fraction of each institution's development work is contributed back to the community, then the whole system gets better and everyone benefits."
- Nothing in it for the software developer? – On the contrary, "Direct participation in an international open source software development project has tremendous advantages for a software developer" in terms of peer-recognition for valuable contributions of code. [2]

3.10.3 Reasons for using OSS

Smart's report concludes:

> There are a number of reasons why some institutions are turning to open source learning environments, but scalability and flexibility are particularly important. Scalability because these open source environments allow institutions to have as many users as they like without incurring bigger licence fees; and flexibility because institutions can choose to develop the open source environment to meet their particular needs.[3]

In addition, according to James Dalziel, designer of LAMS described above, some of the reasons for using the OSS/FS licensing model are as follows[4]:

- Public Good for Education - So… by making LAMS freely available as OSS, potential for widespread use without the barrier of licence fees to limit trials and adoption - Benefits of widespread adoption to education and society far outweigh commercial benefits to a few from limited adoption using licence fees

- Business Reasons - Growing number of viable services businesses built on e.g. technical support or training.

[1] JISC e-Learning Focus, Choosing Open Source Solutions, Christina Smart (July 2005) http://www.elearning.ac.uk/features/oss
[2] JISC e-Learning Focus, Choosing Open Source Solutions, Christina Smart (July 2005) http://www.elearning.ac.uk/features/oss
[3] JISC e-Learning Focus, Choosing Open Source Solutions, Christina Smart (July 2005) http://www.elearning.ac.uk/features/oss
[4] Dalziel, J. 'Case Study: Why did LAMS go Open Source?' presentation May 2005 http://www.melcoe.mq.edu.au/documents/OSSW_03James.ppt

- Enabling Open Standards - By releasing all of LAMS as OSS, anyone can see how the "machinery" of LAMS works, and this will hopefully inform ongoing standards development e.g. development of a Learning Design "tools interface"

- Software Development Reasons - Open Source development processes can produce robust, sophisticated, secure software despite distributed and often voluntary nature of development - In LAMS case, we felt this would be valuable to development, especially given the complexity of LAMS

- Reputational Benefit Reasons – Institutions developing such products gain from being associated with successful OSS projects

- Software Freedom Reasons - Given the potential of LAMS to the future of education, having all software, as well as all data formats, made open encourages freedom in use and development - Choice of the GNU GPL ensures any new development of LAMS by others which is distributed must be shared back with the community on the same licence terms - OSS limits many potential adverse impact on end users of proprietary software'[1]

Finally, according to one review[2], the recent "failure of the UKeU is a good example of the need for collaborative open source efforts in UK e-learning. The practicalities of building a system which is all-encompassing are too complex. A scalable technology infrastructure which can grow as student numbers (and revenue) increase is clearly what is required."

[1] Dalziel, J. 'Case Study: Why did LAMS go Open Source?' presentation May 2005 http://www.melcoe.mq.edu.au/documents/OSSW_03James.ppt
[2] TALL Desk Study available from: http://www.elframework.org/projects/tall/uk_euniversity/

4. Issues with Using Open Source Software

4.1 Introduction

The following review of issues and arguments considers a wide range of questions that are often raised in conjunction with using OSS/FS.

The fact that there are such a range of questions might be an indication that the world of proprietary software now considers OSS/FS to be a significant challenger and therefore needs to be undermined with broad-ranging "issues". As a major player in the proprietary software market, Microsoft has been treated as the bête noire of the Open Source movement for many years. The leaking of the 'Halloween Documents' was hailed as a major coup for the Open Source movement.

> The Halloween documents is the name used outside Microsoft for a series of confidential memoranda on potential strategies related to Open source software and to Linux in particular. The first Halloween document, written by Microsoft engineer Vinod Valloppillil, was leaked to Eric S. Raymond in October 1998…..These documents acknowledged that Open Source/Free Software/Linux products were technologically competitive with some of Microsoft's products, and set out a strategy to combat them. The documents were embarrassing largely because they contradicted Microsoft's public pronouncements on the subject.[1]

One of the main sources for this section of the report has been the influential paper by Wheeler (2005)[2]. This paper provides quantitative data showing that using OSS/FS is often a reasonable or even superior approach to software development compared to using proprietary software. The author stresses that readers should consider using OSS/FS when acquiring software based on such empirical evidence.

4.2 Attitudes, Arguments and Evidence

With regards to the arguments for or against the use of OSS/FS, proponents of both approaches are generally good at supporting their position with favourable evidence.

One of the main "issues" with using OSS/FS is that it seems to contradict the current dominant economic model of advanced capitalism. How can it be possible, many will ask, to produce reliable, well-supported software for free? Wheeler (2005) declares that there has been an attempt:

> to label OSS/FS as "communistic" or "socialistic" (i.e., anti-capitalist), but that rhetoric has failed.[3]

Suppliers of such rhetoric include employees of the Microsoft Corporation. If capitalism is assumed to rely on the concepts of private property then all OSS/FS licensed under GPL copyright is still privately-owned and is thus no threat to the concept of private property.

This issue reveals a lack of information and understanding concerning the true nature of OSS/FS. Since people have much more experience of procuring proprietary software compared with OSS/FS then such misunderstandings can be expected. The DTI (Department of Trade and Industry) in the UK regards some of these ideas as "myths" e.g. that software developed using the OSS/FS model is unreliable compared with the alternative, proprietary model.

---

[1] Taken from Wikipedia and available from: http://en.wikipedia.org/wiki/Halloween_documents

[2] Wheeler, D.A. 'Why Open Source Software / Free Software (OSS/FS, FOSS, or FLOSS)? Look at the Numbers!' revised as of May 9th 2005 available at :http://www.dwheeler.com/oss_fs_why.html

[3] Wheeler, D.A. 'Why Open Source Software / Free Software (OSS/FS, FOSS, or FLOSS)? Look at the Numbers!' revised as of May 9th 2005 available at :http://www.dwheeler.com/oss_fs_why.html

When comparing one software development model against another there is inevitably a mix of quantitative and non-quantitative issues to consider. This section looks briefly at most of the familiar issues.

According to Wheeler there is plenty of quantitative evidence now to support the idea that OSS/FS is at least as reliable and well-supported as proprietary software.

4.2.1 Quantitative issues

Wheeler examines a number of quantitative measures such as experiments and market studies to justify his opinion that:

> using OSS/FS products is in many circumstances a reasonable or even superior approach.[1]

Only market share is presented here to give an example of the evidence, but refer to Wheeler's article for further details.

Although the argument 'that a product is only a winner if it has significant market share' is given short-shrift by Wheeler, the fact remains that OSS/FS now has significant market share in a number of important markets

a) Apache is the leading web server according to Netcraft's statistics
b) GNU/Linux is the leading server Operating System (OS) on the public Internet
c) Sendmail, an OSS/FS program, is the leading email server
d) PHP and MySQL are leading scripting languages and database server technologies
e) Internet Explorer is losing market share to OSS/FS web browsers (e.g. Mozilla-based)

Reliability, performance, scalability, security and total cost of ownership data is also presented by Wheeler to support his assertion that OSS/FS is usually just as competitive as proprietary software in these areas. These issues are examined more closely below.

4.2.2 Qualitative issues

However, it seems that certain qualitative issues are of equal importance. These include:

a) Innovation
b) Protection from licensing litigation
c) Flexibility
d) Freedom from control by another (especially a single source)
e) Social / moral / ethical issues

Some of these issues will be more important to certain parties than others: the Free Software Foundation has a set of papers[2] describing their philosophy, i.e., why they believe Free Software (FS) is an ethical imperative. Active supporters of FS clearly regard social, moral and ethical issues to be highly important. The first three issues are considered to be more relevant to the JORUM service than the others and are further explained here.

a) Innovation

From the point of view of the JORUM service, it could be thought that one of the most important issues would concern **innovation**. Considering that learning object repository software and the supporting technology is still in its early stages of rapid development is important if the needs of users are to be fully met.

---

[1] Wheeler, D.A. 'Why Open Source Software / Free Software (OSS/FS, FOSS, or FLOSS)? Look at the Numbers!' revised as of May 9th 2005 available at :http://www.dwheeler.com/oss_fs_why.html
[2] 'Philosophy of the GNU Project' article available from: http://www.gnu.org/philosophy/philosophy.html

Some people might believe that the proprietary model is the best for encouraging innovation; however, there are arguments to support the view that OSS/FS is better for doing this. For example, Tim Berners-Lee, inventor of the World Wide Web, stated in December 2001 that

> A very significant factor [in widening the Web's use beyond scientific research] was that the software was all (what we now call) open source. It spread fast, and could be improved fast - and it could be installed within government and large industry without having to go through a procurement process.[1]

Wheeler quotes several other examples, making a strong case for this point of view.

The point is that such non-quantitative issues cannot act either for or against using OSS/FS as a general argument: in the main, they simply serve to diffuse any potential worries about OSS/FS being particularly anti-innovation, as critics of this model would tend to argue. In specific instances of using OSS/FS then innovation may either be improved or not, largely dependent on the particular circumstances encountered.

b) Protection from licensing litigation

Institutional procurement of software licences has become a complex matter; there are many licences to keep track of, with the possibility of litigation against licence infringement an expensive risk. Using OSS/FS seems to negate such worries; there is no fear of litigation from the use and copying of OSS/FS.

> Licensing issues do come up when OSS/FS software is modified and then redistributed, but to be fair, proprietary software essentially forbids this action (so it's a completely new right). Even in this circumstance, redistributing modified OSS/FS software generally requires following only a few simple rules (depending on the licence), such as giving credit to previous developers and releasing modifications under the same licence as the original program. [2]

c) Flexibility

Another interesting, and potentially very important issue for JORUM, is that:

> OSS/FS has greater flexibility. OSS/FS users can tailor the product as necessary to meet their needs in ways not possible without source code. Users can tailor the product themselves, or hire whoever they think can solve the problem (including the original developer). [3]

The specific circumstances of the relationship with the software provider may or may not encourage innovation and flexibility of development. Using OSS/FS for the repository would not necessarily make it any more or less easy to develop the software, solve bugs etc.

4.2.3 Some possible fears

Wheeler states there are a mixture of realistic concerns about using OSS/FS and there are many unnecessary fears. The realistic fears include:

- Support

[1] Berners-Lee, T. cited in the article 'Charting the Web's next transformation' by Festa, P. 2001 available from: http://news.cnet.com/news/0-1014-201-8155733-0.html
[2] Wheeler, D.A. 'Why Open Source Software / Free Software (OSS/FS, FOSS, or FLOSS)? Look at the Numbers!' revised as of May 9th 2005 available at :http://www.dwheeler.com/oss_fs_why.html
[3] Wheeler, D.A. 'Why Open Source Software / Free Software (OSS/FS, FOSS, or FLOSS)? Look at the Numbers!' revised as of May 9th 2005 available at :http://www.dwheeler.com/oss_fs_why.html

- Reliability/Performance
- Total Cost of Ownership
- Standards implementation
- Scalability
- Viability

each of which are discussed in more detail below.

The unnecessary fears include:

> support, legal rights, copyright infringement, abandonment, licence unenforceability, GPL "infection", economic non-viability, starving programmers (i.e., the rising commercialization of OSS/FS), compatibility with capitalism, elimination of competition, elimination of "intellectual property", unavailability of software, importance of source code access, an anti-Microsoft campaign, and what's the catch. [1]

The issue of support is discussed in more detail below (see section 3.2.1) but note that Wheeler states that proprietary software is not fundamentally better supported than OSS/FS.

With regards to the legal considerations raised, Wheeler poses two questions:

> **Does proprietary software give users more legal rights than OSS/FS? Or, isn't OSS/FS legally more risky? No** With OSS/FS you give up your right to sue if things go wrong … The obvious retort is that essentially all proprietary software licences *also* forbid lawsuits - so this isn't different at all! Anyone who thinks that they can sue Microsoft or other shrink-wrap proprietary vendors when things go wrong is simply fooling themselves. In any case, most users aren't interested in suing vendors - they want working systems.
>
> **Does OSS/FS expose you to greater risk of abandonment? No.** Businesses go out of business, and individuals lose interest in products, in both the proprietary and OSS/FS world. A major difference, however, is that all OSS/FS programs are automatically in escrow - that is, if their original developer stops supporting the product, any person or group can step forward to support it instead. This has been repeatedly demonstrated in OSS/FS. [2]

One high profile example he quotes is the case of Apache, where NCSA abandoned the product and some of its users banded together to maintain it.

4.3 Drivers for Using OSS/FS

What is the motivation for using OSS/FS when there often seem to be satisfactory proprietary software solutions available?
There appear to be two main driving forces that encourage people to think of using OSS/FS:

- general disappointment with proprietary software solutions
- perceived attractions or benefits of using an OSS/FS development model

These act in a complementary manner.

One of the main disadvantages of using proprietary software solutions is that of cost: very often, software projects can be expensive, considered over the medium to longer-term period. Initial costs include purchasing the licence(s) needed to run the software, as well as continued maintenance of the installation.

---

[1] Wheeler, D.A. 'Why Open Source Software / Free Software (OSS/FS, FOSS, or FLOSS)? Look at the Numbers!' revised as of May 9th 2005 available at :http://www.dwheeler.com/oss_fs_why.html
[2] Wheeler, D.A. 'Why Open Source Software / Free Software (OSS/FS, FOSS, or FLOSS)? Look at the Numbers!' revised as of May 9th 2005 available at :http://www.dwheeler.com/oss_fs_why.html

One of the main attractions of using OSS/FS solutions is the apparent lack of cost – at least where licences are concerned. Also, the right to modify the code can appear to attract certain users of the software. Other factors included the need to have secure and reliable systems, and to be relatively free from the sales-pitch that often surrounds the procurement of proprietary systems.

4.3.1 International Government Policies

The Action Plan (June 2002) for the EC initiative "*eEurope 2005: An Information Society for all*"[1] builds on previous Action Plans setting itself the target "to promote the use of open source software in the public sector …across the Union."

Many governments are interested in the idea of using OSS/FS for a variety of reasons, including the two general drivers stated above.

Mark Bressers, speaking at the "OSS Watch conference: national frameworks" in London on 20th January 2005, said that the main reasons for using OSS/FS in Holland appeared to be:

- 'interoperability - students needing to work in different universities would benefit
- freedom of choice - platform independence greatly increased by using OSS
- enhancing student's software development skills - participation in OSS development is a good inroad for programmers who otherwise would need to be employed by externally-based IT companies. It was noted that, of the top 200 IT software companies in the world almost all of them are US based.' [2]

At the same event Rishab Ghosh presented results from the FLOSS survey (FLOSSPols[3]) indicating the widespread international interest in using OSS/FS. In the Higher Education sectors surveyed, four main drivers for using FLOSS were:

- as public organisations HEIs are themselves influenced by government policies re: FLOSS
- HEIs adopt FLOSS to stretch out their research grant money
- increases the scope for institutional collaborations
- OSS improves skills development in students so this is an obvious choice for HEIs.[4]

4.3.2 UK Government Policy

The UK Government's Cabinet Office[5] officially supports such EC initiatives as *eEurope 2005: An Information Society for all* on OSS/FS by mandating open standards and specifications in its e-Government Interoperability Framework (eGIF). This includes all publicly-funded bodies e.g. the education sector.

The OSS policy of the UK government made certain key decisions, as follows:

- UK Government will consider OSS solutions alongside proprietary ones in IT procurements. Contracts will be awarded on a value for money basis
- UK Government will only use products for interoperability that support open standards and specifications in all future IT developments
- UK Government will seek to avoid lock-in to proprietary IT products and services
- UK Government will consider obtaining full rights to bespoke software code or customisations of COTS (Commercial Off The Shelf) software it procures wherever this achieves best value for money

[1] eEurope 2005 Action Plan at http://europa.eu.int/information_society/eeurope/2005/all_about/action_plan/index_en.htm
[2] The JISC OSS-Watch conference 2005 http://www.oss-watch.ac.uk/events/2005-01-20/
[3] FLOSS survey available from: http://flosspols.org
[4] The JISC OSS-Watch conference 2005 http://www.oss-watch.ac.uk/events/2005-01-20/
[5] Cabinet Office document 'OPEN SOURCE SOFTWARE Use within UK Government' Oct 2004 available from: http://www.govtalk.gov.uk/documents/oss_policy_version2.pdf

Publicly funded R&D projects which aim to produce software outputs shall specify a proposed software exploitation route at the start of the project. At the completion of the project, the software shall be exploited either commercially or within an academic community or as OSS.

Again, the Department of Trade and Industry in the UK published a fact sheet extolling the virtues of using OSS/FS to business, stating that using OSS/FS increases

> choice in the software market … helping businesses to cut their IT costs [1]

The UK Office of Government Commerce (OGC) that works with government to improve procurement and project/programme management reported on "Proof of Concept" trials of Open Source Software (OSS) in a range of public bodies in conjunction with IBM, and later with Sun Microsystems. In their Final Report on the trials the general conclusions were that OSS:

- [was] a viable and credible alternative to proprietary software for infrastructure implementations
- can generate significant savings in hardware and software costs for infrastructure implementation[2]

Some of the recommendations from the OGC were that:

> In the light of the outcomes of the Proof of Concept trials … public sector bodies should:
>
> examine carefully the technical and business case for implementation of Open Source software and the role which OSS could play in current and future projects, working with their outsourced IT providers where appropriate
> identify the role of open standards in future IS/IT strategy and policy, in conformance with the e-Government Interoperability Framework (eGIF) [3]

One of the main justifications for general UK Government policy on software procurement is to use a solution that gives value for money. In practice this may be an OSS/FS solution, or a proprietary one, or a mixture of both. Decisions are to be made on a case-by-case basis.

Every effort is to be taken to reduce the cost and risk to government-funded systems and to provide more flexibility in the development and enhancement of such systems, "vesting the ownership of bespoke and tailored software code with Government where this offers value for money".

4.3.3 JISC Policy on OSS/FS

In the UK, the JISC have investigated and supported various software projects whose outputs have been released under OSS/FS licence.

The JISC have a draft policy about using OSS/FS and its aims include the need to:

- ensure a good return on the investment of public funds
- improve the sustainability of fixed-term funded projects/systems over the longer-term
- use OSS to enable better sharing of software in collaborative projects with international partners which the use of proprietary software makes difficult[4]

[1] Stead, J. 'OPEN SOURCE SOFTWARE' DTI Factsheet Oct 2004 available from: http://www.dti.gov.uk/bestpractice/assets/oss.pdf
[2] 'Open Source Software Trials in Government Final Report', Oct 2004, Office of Government Commerce available from: http://www.ogc.gov.uk/embedded_object.asp?docid=1002367
[3] 'Open Source Software Trials in Government Final Report', Oct 2004, Office of Government Commerce available from: http://www.ogc.gov.uk/embedded_object.asp?docid=1002367
[4] JISC draft policy on OSS/FS, personal communication from the OSS-Watch, Jan 2005

The Guidelines for all JISC services and projects including both development activities and services providing advice and guidance to the higher and further education communities should:

Advice and guidance to the communities JISC serves must be neutral and unbiased, and must not discriminate between open source and closed source software products.

For bodies awarding JISC funding:

Calls for funding, the bidding process, the award of funding, and the administration of awarded funding and the evaluation of funded projects and services must not discriminate between open source and closed source software, unless the purpose of the projects or services specifically requires it.

Where open source and closed source software are evaluated against one another, value for money over the expected lifetime of the system must be compared.

For any new projects specifically funded by the JISC to produce software:

Copyright of software, documentation, design materials, manuals, user interface and source code must be released under an OSI-approved open source licence, unless the bid explicitly argues why this should not be the case and proposes an alternative licence.
Software must in any case be licensed and publicly available, for any use and at no financial cost, throughout UK higher and further education
Patents are the alternative means of asserting ownership of intellectual property in software. Although the JISC will make no claim to patents if a project – or staff working for a project – wishes to apply for a patent, the project must ensure that the existence of the patent in no way interferes with software use, modification or re-distribution under the chosen software licence and that the software is licensed and publicly available, at no financial cost, throughout UK higher and further education

The JISC collaborates on the e-Learning Framework[1] and through the JISC e-Learning Programme considers issues of Open Specifications and Standards. As part of the tools strand of the e-Learning Programme certain software tools have been developed according to the OSS/FS development model.

That the JISC is interested in the issue of investigating developments in OSS/FS is clear from their funding the OSS Watch Service based at Oxford University.

Also, a project team from the University of Strathclyde and Kilmarnock College has produced a guide and a decision-making framework to help managers in HE/FE select software and services from proprietary, in-house or OSS[2].

4.4 Main Issues

There are widely divided opinions about the desirability of using OSS/FS in general: some people are evangelical about OSS/FS and others are very sceptical.

Proponents of OSS/FS have claimed that proprietary software is expensive, restrictive, inflexible and so on and cite certain monolithic software companies (such as Microsoft) as being the villain of the piece with their widely-publicised monopolistic practices and frequently bug-ridden software.

[1] E-Learning Framework http://www.elframework.org/
[2] Coen, M. et al 'A Guide to Investing in Proprietary, In-House or Open Source Software and Services' Jan 2005 available from: http://www.predict.strath.ac.uk/projects/sw_services/

Sceptics have countered such views by saying that OSS/FS is unreliable, and not yet a realistic alternative to proprietary systems. A DTI fact sheet refers to such ideas as myths[1].

The fact remains that there are issues associated with any large-scale software development, regardless of whether it is OSS/FS or proprietary. In addition, there are other questions asked about OSS/FS in the context of its alleged benefits and drawbacks.

These questions concern:

- Support
- Reliability/Performance
- Total Cost of Ownership
- Standards implementation
- Scalability
- Viability

4.4.1 Support

One of the main arguments against using OSS/FS is the issue of support.

The concern here is that, if the software is free, then the support is likely to be absent. Many people believe the old adage that you get what you pay for. In the world of OSS/FS this is not necessarily true.

In June 2004 the OSS-Watch conference specifically examined "Support Models for Open Source Deployment and Development". Commenting on some problems for OSS/FS Roy Lane of Oracle, noted that:

> support is informal; there is no roadmap for change; there are functional gaps in the software; and that licensing is problematic and so on

Sebastian Rahtz of OSS Watch noted that these problems can often be turned into assets; namely that, regarding support there are a number of possible solutions, such as:

- **DIY** – Do IT Yourself, as for example, the Bodington VLE project team have done, by providing in-house support, where the In-house staff gain development skills in maintaining and developing the software. The Open University has invested in an eProduction System, using some OSS/FS.

- **Join a consortium** – e.g. the uPortal project where the University of Hull joined prestigious partners such as Yale in the US. The advantages of joining a consortium include access to a genuine community of users of the software, sharing experiences amongst respected partner institutions in this case.

- **Employ a specialist consultant** – an independent consultancy, OS Consult, provide a range of support services to the makers of Apache Lenya, an open source content management system based on Apache Cocoon. OS Consult offer these support options for OSS/FS:
  Consulting, design, development, installation
  Outsourcing
  Per incident support
  Longer term support contracts
  Training suited to your requirements

  One example consultancy, Parthenon Computing Ltd, provide:
  Expertise - consultants are often the founders and long-term contributors to open source projects.
  Flexibility - as much or as little consultancy resource when needed.

[1] OPEN SOURCE SOFTWARE http://www.dti.gov.uk/bestpractice/assets/oss.pdf

Team - a team with mixed roles is better than a multi-talented individual.
Value - expertise, flexibility and a strong team represent good value

- **Using existing vendors** - Novell can support the implementation of OSS within higher education by providing commercial support. Novell have made a long term commitment to open source and that they are migrating the full range of their own systems to OSS right through from their back end servers through to desktop machines and software such as OpenOffice.

  Sun Microsystems is a prominent commercial supplier of computer hardware and software and they are big supporters of OSS/FS. Through the Java Education and Learning Community (now the GELC) members can

  > ..find, develop and share Java-related open source educational tools, open learning standards implementations, and open course learning materials.[1]

  GELC also have a community portal running discussion forums and mailing lists, hold a number of events and publish a number of white papers and newsletters.

The role of the community is very important in the development, and subsequent support of OSS/FS. This is largely due to factors such as the enthusiasm of a highly dispersed group of software developers who historically led the way in OSS/FS development (the Linux route) and to the radically different management model used, compared with proprietary, closed software providers. This model is described in the article “The Cathedral and the Bazaar” by Eric Raymond[2].

4.4.2 Reliability/Performance

Reliability of software developed through OSS/FS is said to be high (and sometimes higher) than closed, proprietary systems. Wheeler (op cit) states that the Linux Operating System is often claimed to be more reliable than MS Windows but he also makes it clear that reliability is difficult to properly assess since many complex factors need to be taken into consideration.

The fact that some governments around the world trust their mission-critical software systems to OSS/FS indicates increasing confidence in OSS/FS. The US Department of Defence helped develop a specially-secure version of Linux (Security Enhanced Linux (SELinux) for its own systems.

However, as a counter argument, Viega, author of GNU Mailman software, states:

> it seems the availability of source code does not automatically guarantee that the code has been reviewed by competent parties for a variety of reasons. Secondly, people who are looking at the source code with the intent of modifying it are not necessarily in the state of mind to perform a comprehensive security audit of the code.[3]

Viega’s views on security and OSS/FS act as an antidote to the hype ("Given enough eyeballs, all bugs are shallow") issued by OSS/FS evangelists such as Eric Raymond writing in his article “The Cathedral and the Bazaar”.

> One deterrent to the mass review of certain Open Source projects is a high level of complexity in the code, which can be compounded by a lack of documentation … Another reason that prevents good review of Open Source code is that most of the people only look at the parts of the code that they want to modify which may only be a small section of the code. This behaviour leads to various "hotspots" in the code that are intensely reviewed because they are the most open to modification

[1] Java Education and Learning Community, http://www.sun.com/products-n-solutions/edu/feature/jelc.html
[2] Eric Raymond, The Cathedral and the Bazaar, available from: http://www.tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html
[3] John Viega, The Myth of Open Source Security , available from: http://www.earthweb.com/article/0,,10455_626641_1,00.html

> while many other sections of the code that are less likely to be useful during modifications are barely looked at. [1]

4.4.3 Total Cost of Ownership (TCO)

In large software projects the cost of licensing software can be considerable, amounting to several thousand pounds per annum. Other costs can also be high, such as on-going support contracts with the supplier, or other agency. OSS/FS is usually free to licence although this is not always so (since OSS/FS is NOT "free" in the sense of free beer).

As there are few (if any) OSS/FS Learning Object Repositories currently able to fully meet the requirements of JORUM then it is not easy to make a real cost comparison between the possible OSS/FS products and commercial solutions.

In contrast, it **is** possible to compare a well-known OSS/FS product (e.g. some version of the OSS/FS Linux) with a comparable version of the proprietary MS Windows (or other Operating System). Even then, it is necessary to make a number of assumptions. There are various "hidden" expenses, such as administration costs, upgrade costs, technical support, end-user operation costs, and so on.

According to Wheeler

> the term "TCO" is common but misleading for most software, *especially* for proprietary software, because software users often don't own the software they use and thus don't have the rights of ownership. It might be more accurate to say that proprietary software users often "lease" or "rent" the software, and thus this category could more accurately be named "total cost to lease or own". Fundamentally, unless you arrange to have a software program's copyright transferred to you, you do not actually own the software -- you only own a *licence* to *run* the software in certain limited ways. That's an important distinction; in particular, with proprietary software you typically *do not have* the rights associated with ownership.
>
> [in contrast] the rights OSS/FS users are granted (users can understand, publicly comment on, modify, and redistribute the software -- and all this in perpetuity) are far closer to an owner's rights than the rights granted to a proprietary software user.[2]

With proprietary software there are often recurring costs (e.g. to upgrade, or to remain compatible with other people's software).

In short, the cost comparisons between OSS/FS and proprietary software are far from being straightforward, and can be very misleading.

For a niche-market application such as a Learning Object Repository it is possible that there is little cost difference between the two types of solution: however, on purely TCO grounds certain costs are likely to be absent from the OSS/FS solution (e.g. recurrent annual licence fees). However, ongoing support for the development and maintenance of the software will exist for both models, OSS/FS or proprietary.

In conclusion, despite initial assumptions, cost factors are not the main reason for choosing the OSS/FS route.

---

[1] Eric Raymond, The Cathedral and the Bazaar, available from: http://www.tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html

[2] Wheeler, D.A. 'Why Open Source Software / Free Software (OSS/FS, FOSS, or FLOSS)? Look at the Numbers!' revised as of May 9th 2005 available at :http://www.dwheeler.com/oss_fs_why.html

4.4.4 Standards Implementation

Although the technical issues around Open Standards and Specifications is discussed in detail in section 5.4.3 below some comments about standards implementation are made in this context.

Standards implementation is one of the key issues as far as Government software procurement policy is concerned and is also of high priority for the JORUM project. In general it is important for procurers of new software to know that it will work with existing software and hardware they are using. They are also likely to want the data created in one application to be exchangeable with data from others. This is the issue of interoperability, which compliance with existing standards and specifications is intended to ensure.

As far as the JORUM service is concerned, one of the most important specifications governing choice of repository system is the IMS Content Packaging specification[1]. This specification concerns the sending of learning resources (or learning objects) from one program to another, thus facilitating easier delivery, reuse and sharing of materials. Amongst other purposes the IMS Content Packaging specification is for:

Transmission of content from one VLE to another
Transfer of content from an authoring tool to a VLE or repository
Assembly, or aggregation of existing content into a single package for delivery

This specification is particularly useful in the context of an e-Learning programme which needs the various component tools to be able to exchange learning content with each other. Although there are several Institutional Repositories available at the moment few actually implement this specification (including the OSS/FS DSpace repository) at this point in time (see 5.2.1 for further details).

According to Brian Kelly (UKOLN) the issue of Open Standards may be more important than whether the system is Open Source or not[2]. Open Standards are critical for the development of open, cross-platform, vendor neutral services providing application and device independence, ensuring long term and wide accessibility. Kelly also points out that Open Source software may have a role to play in supporting Open Standards, but proprietary software can also be used with Open Standards. Indeed, the current commercial supplier of repository software to JORUM, Intrallect Ltd, is based on open standards.

4.4.5 Scalability

Defining scalability (regardless of the particular software) as being able to move up from a modest installation to larger ones, then, according to Wheeler, the scalability of many OSS/FS products is “amazing” (mainly due to the openness of the source code, and the ability of contributors to identify potential scalability problems).[3] Certainly, the GNU/Linux operating system software works on an impressive range of hardware systems from the individual PC system to large mainframes. Scalability, then, need not be a particular problem for OSS/FS.

4.4.6 Viability

As far as the JORUM repository is concerned viable could mean “capable of growing or developing”. This idea is important for the repository regardless of whether or not a proprietary or OSS/FS solution is chosen at some future date. User needs typically change over time, so the capacity to easily modify the software is important.

An OSS/FS solution is perhaps as easily adapted to incorporate changing functionality as is a proprietary solution. The Linux experience convinced people (e.g. Eric Raymond) that OSS/FS

[1] CETIS briefing document available from: http://www.jiscinfonet.ac.uk/Resources/external-resources/CPbrief.pdf
[2] Available from: http://www.oss-watch.ac.uk/events/2003-12-11/open-standards.pdf
[3] Wheeler, D.A. ‘Why Open Source Software / Free Software (OSS/FS, FOSS, or FLOSS)? Look at the Numbers!’ revised as of May 9th 2005 available at :http://www.dwheeler.com/oss_fs_why.html

can evolve new code very rapidly, even to the extent of developing a full operating system consisting of millions of lines of code.

Considering the full range of factors (technical and organisational) important to the development of something like the JORUM repository, then there is no prima facie reason why OSS/FS cannot be used as a software solution. According to Government guidelines regarding new software procurement these things need to be judged on a case-by-case basis.

4.5 Summary of Advantages and Disadvantages of Using OSS/FS

Although the issues are complex it is possible to list the most prominent general advantages and disadvantages in the choice of software development models available.

4.5.1 Advantages

- **Reduced costs** due to OSS/FS licensing model; less likely to incur recurrent licensing costs compared with proprietary systems
- **Greater control of the source code**, bringing more flexibility to add necessary modifications to further development. Less fear of vendor tie-in.
- **Wider community** able to contribute to development of source code, to quickly evolve the code and fix bugs
- **Skills development** within the UK HE and FE community would be enhanced in the continued development of OSS/FS tools for e-Learning that contributes to the creation of an e-Learning Framework
- Possibly **greater reliability and scalability** of the whole project, based on a modular approach using OSS/FS tools already developed by the community

4.5.2 Disadvantages

- **Licensing models** for OSS/FS are unclear to most managers in the HE and FE sectors with considerable confusion existing about the meaning of OSS/FS in this context
- **Support models** are often perceived to be problematic where a different model of support may need to be used compared with more familiar proprietary software procurements
- **Management** of OSS/FS means that a project to develop a LOR for the UK HE/FE sector would need to be well-focussed and committed. The experience of the wider OSS/FS community in such project management should act as a guide
- **Developing software such as a LOR from scratch** would be daunting for most projects; some projects have foundered due to such problems (e.g. BCcampus decided not to use APOLLO since OSS/FS development proved more to be more problematic than they anticipated (see Appendix).

5. Digital Repository Technology

5.1 Definitions

The following definitions highlight the distinction between Learning Object Repositories (which are the principal focus of this section of the report) and more generic digital repositories.

5.1.1 Generic Digital Repositories

There are several types of repository systems that function in a similar fashion in that they store digital content, and have some means of indexing such content (using various metadata schemas). Many of the repository systems emerging from the library world are of this type. However, they are not exactly what the JORUM project requires since they do not always comply with the required standards and are not necessarily designed to cope with LOs. One of the key features missing from such systems is the ability to import/export IMS Content Packages.

These generic repositories are often implemented as **Institutional Repositories** – which usually means that they address the specific needs of the Institution in question (e.g. to store various types of scholarly output). The EPrints[1] software is of this type and has one of the largest installed user bases in the world.

The generic digital repositories are more mature products than the relatively newer, learning object repositories.

Section 6 considers how some digital repositories are being developed for use with LOs.

5.1.2 Learning Object Repositories

The JORUM project's working definition of a Learning Object Repository (LOR) is as follows:

> A learning object repository (LOR) is a collection of LOs having detailed information (metadata) about them that is accessible via a network or the Internet. In addition to housing LOs, repositories can store 'locations' for objects that are held elsewhere.[2]

This definition includes the ability to store 'locations' for objects that are held elsewhere – in other words, Virtual Objects or URLs. This widens the range of the LORs content to include material stored on distributed servers almost anywhere on the Internet. The LOR is not simply a database of LOs viewed as a silo – or warehouse – of learning materials.

Although the definition makes no reference to the ability to import/export Content Packages this is a key requirement of the JORUM service repository to facilitate interoperability between e-Learning systems in the UK setting of the JISC Information Environment. The following two paragraphs recap the distinctions drawn in a previous JORUM report.[2]

5.2 OSS/FS Generic Digital Repository Systems

As mentioned above, there are many more generic digital repositories being developed at the moment (often used as institutional repositories) than there are learning object repositories (LOR).

The following provides a brief description of some OSS/FS institutional repositories that may conceivably move towards a LOR type of functionality in the future: DSpace, Fedora, EPrints.

[1] http://software.eprints.org/press/#name
[2] Volume II - Review of Software Systems available from: http://www.jorum.ac.uk/research/archive/docs/vol2_Fin_abridged.pdf

5.2.1 DSpace

This is an open source digital repository system from the MIT Libraries and Hewlett-Packard Labs and is mainly an example of archiving software (e-prints and other scholarly artefacts). DSpace[1] is one of the leading developers of OSS in the repository field that captures, stores, indexes, preserves, and redistributes an organisation's research data. The emphasis is definitely on research output from the participating institution. Since February 2002 DSpace source code has been available from SourceForge as Open Source software. There is considerable support for its future development, as judged from the existence of Wiki etc. The large user base amongst US universities is creating a real community from which the DSpace developers can learn[2] via feedback. Although a repository system such as DSpace is not currently, in its default installation, a LOR that features IMS Content Packaging, later versions may include this functionality.

Internationally, there are many institutions[3] developing repositories based on DSpace ; two examples are the Bristol Repository of Scholarly EPrints (ROSE)[4] in the UK and the Brigham Young University[5] in the US. Another example is the UK based Open Repository[6] that is a service from BioMed Central for institutions and organizations to build and maintain repositories on their behalf and uses OAI (Open Archives Initiative) protocol for metadata searches. BioMed Central will set up and install the entire repository quickly with proven, well-supported software.

Some of the main technical features of DSpace are: that installations are typically Unix-based, with the code being written in Java; there is a web interface; the metadata is Qualified Dublin Core and is OAI-PMH compatible; retrieval of resources uses standard search engine techniques; like Fedora, there is guaranteed bitstream preservation and persistent identifiers ("handles") are provided for resources within the repository; all file formats are accepted, but some offer better preservation possibilities.

5.2.2 Fedora

Fedora (Flexible Extensible Digital Object Repository Architecture)[7] began as a DARPA and NSF funded research project at Cornell University, where the initial reference implementation was developed by Payette, Lagoze et al. By 2002 full-scale development had started with grant funding from the Andrew W Mellon Foundation. Fedora 1 was released in May 2003 with subsequent releases approximately every quarter, adding functionality and correcting bugs discovered by users and the Fedora development team. In June 2004, the Foundation funded Fedora Phase 2 for an additional 3 year project.

Fedora is a general purpose repository service developed jointly by The University of Virginia Library and Cornell University. The Fedora project is devoted to the goal of providing open-source repository software that can serve as the foundation for many types of information management systems. The Fedora Repository System is free open-source software distributed under the Mozilla Public Licence Version 1.1. It requires the Sun Java Software Development Kit, version 1.4 or higher.

At the time of writing, the Fedora website lists 24 institutions (mainly in HE) using Fedora, mostly in the USA.

The Fedora software demonstrates how distributed digital information management can be deployed using web-based technologies, including XML and web services. The implementation of web services technology seems extensive using REST and SOAP interfaces.

---

[1] DSpace website at http://dspace.org/
[2] http://wiki.dspace.org/LessonsLearned
[3] From the community Wiki (http://wiki.dspace.org/DspaceInstances
[4] http://rose.bristol.ac.uk/dspace/
[5] https://dspace.byu.edu/
[6] http://www.openrepository.com/
[7] http://www.fedora.info/

> Fedora digital objects are managed within the Fedora Service Framework which consists of a set of loosely coupled services that interact and collaborate with each other. At the core of the Fedora Service Framework is the Fedora Repository Service which exposes interfaces for managing and accessing digital objects in a repository. Each service interface is defined using the Web Service Description Language (WSDL).[1]

Technical details of the Fedora Object Model can be found in the DLib Magazine article by Staples et. al., entitled 'The Fedora Project'[2].

There are various Use Case scenarios possible with any institutional repository software and four different scenarios are described in this article relating to Fedora:

Use Case 1: Fedora "out-of-the-box"
Use Case 2: A digital asset management system
Use Case 3: A digital library for a research university
Use Case 4: Fedora for distributed content objects

> The Fedora architecture is based on object models that by definition are templates for units of content, called data objects, which can include digital resources, metadata about the resources, and linkages to software tools and services that have been configured to deliver the content in desired ways.[3]

In the object model used by Fedora the digital resources and the metadata are datastreams; the content of a datastream is identified using a URL. Behaviours act on these units of content.

> These behaviors can be exploited to deliver varieties of prepared content directly to a web browser. They can also be used to prepare or configure content to be used through some external software application. In a sense, these object models can be thought of as containers that give a useful shape to information poured into them; if the information fits the container, it can immediately be used in predefined ways.[4]

5.2.3 EPrints

This software is often used as an Institutional Repository and, according to the OSI Guide (January 2004) EPrints has the largest installed base of any repository system described in their guide[5]. Originally developed at the University of Southampton EPrints provides a web interface for managing, submitting, discovering, and downloading documents. According to the author of the EPrints software, Christopher Gutteridge[6], the software is called EPrints because it is typically used to collect and share eprints (lower case "e"). An eprint is an electronic version of a research paper which is shared online.

EPrints repositories create online archives and are compliant with the Open Archives Initiative (OAI). The default configuration is a repository of the research output of an academic institution but an Eprint archive can be adapted for many more purposes. The UK JISC support EPrints as part of their Open Citation Project.

---

[1] Cover Pages: Fedora Version 2.0 Open-Source Repository Supports XML and Web Services article available at: http://xml.coverpages.org/ni2005-03-18-a.html
[2] DLib Magazine article April 2003 available from: http://www.dlib.org/dlib/april03/staples/04staples.html
[3] DLib article
[4] DLib article
[5] OSI Guide to Institutional Repository Software 3rd August 2004 Edn. available at: http://www.soros.org/openaccess/pdf/OSI_Guide_to_IR_Software_v3.pdf
[6] http://software.eprints.org/press/#name

5.2.4 Summary

There are some prominent developments in OSS/FS Institutional Repositories (IR), in particular DSpace, Fedora and EPrints who have the largest market share at the moment. Other systems (e.g. based on CMS, such as Plone) were included in this section since they are positioned within the learning management arena, in particular Connexions. Several other IR systems that implement the OAI-PMH protocols are listed from the OSI Guide[1].

Although these IRs do not specifically manage learning objects at present it may be that some of them will be adapted to serve as OSS/FS LORs in the future. Similar technical issues (to those of LORs) are being addressed by systems such as DSpace and Fedora in terms of the indexing of objects, design of workflow and so on. They are all attempting to comply with relevant interoperability standards.

However, very few IR systems have as their principal focus the storage and retrieval of 'learning objects' so their immediate relevance to JORUM is still rather limited.

5.3 JORUM Requirements

The requirements for the JORUM service repository system were made clear in the tender documents issued through the EU procurement process.

The JORUM project was tasked to procure a repository system for a JISC-sponsored learning object repository (LOR) service for the whole of the HE/FE sector in the UK. After a thorough evaluation period, JORUM awarded the three-year contract to Intrallect Ltd for the IntraLibrary product. The JORUM service in development will be launched in late 2005, initially for the X4L projects to deposit the outputs from various JISC-funded projects. For more information on the JORUM Service in Development please refer to the website[2].

After an extensive set of scoping studies, producing a series of documents[3], into a range of relevant issues (including metadata, the JISC IE, licensing and workflow) the JORUM project produced a list of requirements for the JORUM LOR. The final specification for the initial version of the system is available on the JORUM project site[4]. The basic technical requirements can be summarised as:

The repository needs to:

- Enable interoperability between services/systems in the JISC IE.
- Be standards based.
- Be platform independent.

The following features/functionality should be available:

- Upload and storage of Learning Objects.
- Preview and download of Learning Objects.
- Role-based authorisation.
- Content Packaging, in accordance with the IMS specification.
- Workflow management.
- Searching of metadata.
- Import of metadata.
- Harvesting of metadata via OAI-PMH.

The JORUM project team has worked closely with the supplier to obtain a customised LOR from that meets the vast majority of their requirements, with further development work planned.

---

[1] OSI Guide to Institutional Repository Software 3rd August 2004 Edn. available at: http://www.soros.org/openaccess/pdf/OSI_Guide_to_IR_Software_v3.pdf

[2] JORUM Service in Development website at www.jorum.ac.uk

[3] See http://www.jorum.ac.uk/research/archive/research/publications.html

[4] JORUM Repository Requirements Specification: http://www.jorum.ac.uk/research/archive/docs/pdf/JorumSpecification.pdf

The amount of work that has been needed to hone these precise requirements for the UK HE sector and implement them in a relatively user-friendly interface should not be underestimated.

It is probable that if OSS/FS were to be used in the future for the JORUM service it would be much easier to make use of the existing development work already carried out by Intrallect Ltd. The alternative would be to build the new repository system from the ground up, either taking existing OSS/FS products and customising them, or to start from the beginning entirely, or a combination of both (new tools may need to yet be developed). Any solution based on OSS/FS would require a significant amount of development and customisation to be carried out in order to meet the present JORUM requirements, not least because the area of LOR development is still relatively immature.

5.4 Relevant Technologies – Open Standards and Specifications

5.4.1 Context

One important aspect of the JORUM repository is that it needs to fit within the developing JISC Information Environment (JISC IE):

> (this) technical architecture specifies a set of standards and protocols that support the development and delivery of an integrated set of networked services that allow the end-user to discover, access**,** use and publish digital and physical resources as part of their learning and research activities.[1]

The key aim of the JISC IE is to facilitate the integration of disparate information resources with the various resource 'discovery' services (e.g. the RDN) and the information resources and learning object repositories with Virtual Learning Environments.

The JORUM repository is seen as being a part of the 'provision layer' of the JISC IE architecture[2]. That is, it will be a content provider to components within the 'presentation layer' of the JISC IE such as the joint JISC and Higher Education Academy[3]-funded Connect services[4], or other systems.

Another important context is the JISC e-Learning Framework (ELF). The e-Learning programme within the UK is led by the HEFCE and JISC whose e-Learning Framework (ELF) takes a particular view of how the various e-Learning tools and services need to fit together. HEFCE believe it is possible to build an open accessible system within a common digital infrastructure (or framework) and their e-Learning strategy is aimed at developing tools that encourage the sharing of resources. Interoperability is at the heart of the ELF, enabling the integration of commercial, home-grown and open source components and applications within institutions.

The JORUM repository will need to fit into the ELF. As part of the JORUM project's continuing research and development work the underlying technology to achieve these goals will be examined in related reports[5]. The IMS General Web Services specification will underpin a Service Oriented Architectures (SOA) approach to system construction. Roughly speaking, this equates to combining loosely-coupled component services or tools to develop local infrastructures for institutions, where the components need not be located on the same host. Each component is the realisation of a service, which may be thought of as the union of one or more interfaces. All communication between components is in the form of message passing.

[1] JISC Information Environment Architecture available from: http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/
[2] JISC Information Environment Architecture available from: http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/
[3] Higher Education Academy http://www.heacademy.ac.uk/
[4] http://www.connect.ac.uk
[5] JORUM report on Technical Frameworks and Infrastructures

5.4.2 Web Technologies

There is a variety of basic web technologies involved in building and operating a digital LOR in the context of other e-Learning software systems currently in use today, and almost all of these technologies are Open Source.

The specifications and standards used to build learning object repository systems that can share data and services between them can be considered as an essential part of the required technology. The JORUM service puts interoperability at the core of the service it is setting up.

The IMS Digital Repositories Interoperability Specification (IMS, 2003)[1] makes recommendations for the interoperation of the most common repository functions (i.e. core functions like Search/Expose; Submit/Store; Request/Deliver; Gather/Expose). These recommendations enable services to provide a common interface. This is termed a 'Functional Architecture'.

The markup language XML is used for storage in some repositories. These repositories may use XQuery, a declarative query language which provides similar search capabilities over collections of XML data to that provided by SQL for relational databases. Other possible solutions for searching XML include XPath and XSLT (eXtensible Style Language Transformations), which currently have more implementations than the relative newcomer XQuery. However, they lack the flexibility and some of the functionality of XQuery.

Other established repository systems (often in the library world) use the Z39.50 protocol.

Since most LOR are accessible over the Web then they will inevitably depend on certain protocols such as HTTP for sending and requesting web pages. The SOAP (Simple Object Access Protocol) uses the W3C standard of XML as a mechanism for exchanging structured information between peers in a distributed system of repositories. Combined with the WSDL (Web Services Description Language) SOAP allows Web service providers to specify what services they can provide, what the inputs and outputs of the services will be and how to encode/decode the requests and responses exchanged between the clients and servers. SOAP and WSDL are neutral as to the programming language used in the system and free toolkits are available in a number of programming languages.

e-Learning applications usually depend on the basic web technologies and standards including the various markup and scripting languages (HTML, XML, PHP, JavaScript, Perl and so on) as well as full programming languages like Java. All of these technologies are used, in various combinations, to build Web applications. Well-known server technologies such as Apache, or Apache Tomcat deliver many web-based applications today.

At the heart of any repository lies database technology. There is a variety of database formats and DBMS on the market but one of the most popular databases used in Web applications is MySQL. Many repositories use MySQL along with scripting languages such as PHP to store the learning objects and associated metadata. MySQL is OSS/FS.

Most of the above technologies are now accepted standards by the Internet community, and various standard-setting bodies e.g. IEEE, IMS Global Learning Consortium, W3C etc make these standards open and publicly known.

In the area of e-Learning, there is considerable value attached to the idea of adopting standards due to previous experiences of being frustrated by incompatible, proprietary standards. In the UK, the JISC-funded CETIS[2] (Centre for Technology and Interoperability Standards) was set up to promote knowledge and awareness about the wide range of standards and specifications relevant to e-Learning.

The IMS have produced a standard for Content Packaging (CP) that is widely adopted by the e-Learning community. Content Packaging concerns the techniques used to bundle together the

[1] IMS Digital Repositories, v.1.0 Final specification, January 30th 2003 available from: http://www.imsglobal.org/digitalrepositories/
[2] CETIS http://www.cetis.ac.uk/

various files used in the creation of online learning materials (e.g. small courses) in a way that makes them easy to import and export into other e-Learning systems (e.g. a Virtual Learning Environment). Many LORs provide this functionality in their systems. Support for CP was a key requirement for the JORUM repository.

One of the most important sets of standards relates to ensuring that systems can make their resources (including LOR) available to others doing cross-searches to find and retrieve those resources. This is the area of Metadata. The IEEE LOM is a learning object metadata standard that is widely used in education, although there are many variations of this standard used in different work contexts. Specific Application Profiles have been developed for these contexts especially concerning e-Learning and LOR.

If LORs are developed in a distributed network (as in the Canadian e-Learning programme) then some way of harvesting the metadata records of all contributing repositories has to be used. The Open Archives Initiative (OAI) has been active in promoting the appropriate protocols to enable this and have developed the OAI protocol for metadata harvesting (OAI-PMH).

To this extent all the necessary technological tools and techniques to build LORs are already Open Source.

5.4.3 Open Standards and Specifications

From the above discussion it should be clear that there are many standards and specifications relevant to the provision of a learning object repository service.

Although the terms standards and specifications are often used interchangeably, according to the JISC Infonet website, they do have a precise meaning:

> Essentially standards are authoritative having been produced by a national or international organisation with the legal power to define standards. Specifications on the other hand have no legal status and may be defined by anyone. Some specifications (including some from IMS) go on to become standards, but that may take many years. For instance, the standard for HTML is 2.0, while the world is using HTML 4.01 with all the additional features that it offers.[1]

Standards can include protocols and file formats that are openly documented, accepted by the industry and freely available for adoption. For the Internet some of the most basic and important standards include HTTP, HTML, WAP, TCP/IP. The following file formats are ubiquitous: PDF, GIF, JPG, MS Word.

There are several standards bodies active in the area of learning technology, e.g. IEEE Learning Standards Committee[2], IMS Global Consortium[3], W3C[4], ISO[5], ECMA[6], IETF[7]. These bodies are typically composed of Government and Industry people acting in consortia. The IT industry supports these standards - for example, IBM is a major contributor and supporter of Open Standards.

5.4.4 What are Open Standards?

Open Standards can be defined as:

> a set of principles and a method of practice for developers of open source software. Roughly speaking open standards require developers to make their material freely available for others to use and implement with no royalty or fee. The

[1] Available from: http://www.jiscinfonet.ac.uk/InfoKits/creating-an-mle/technology-options/standards-and-specifications
[2] http://ltsc.ieee.org/
[3] http://www.imsglobal.org/
[4] http://www.w3.org/
[5] www.iso.org/
[6] http://www.ecma-international.org/
[7] http://www.ietf.org/

intention is for open standards to create a fair market - a customer isn't locked in to a particular software vendor or group.[1]

Some definitions describe an open standard as one which is documented, available for all to use, and free of charge. Others put fewer restrictions on the definition of "open" - the standard must be available but you might have to pay to see it, and it might contain patents, in which case you must pay for using it.[2]

Unfortunately, there is no universally-agreed definition of Open Standards (unlike Open Source), although there is often said to be a close association between the philosophies of Open Source and Open Standards.

According to Brian Kelly of UKOLN[3], the use of open standards seeks to achieve:

- application independence
- device independence
- wide accessibility
- long term access
- neutral ownership & avoidance of licensing conditions, patents
- architectural integrity

Kelly maintains that Open Standards are critical for the development of open, cross-platform, vendor neutral services.

Although some of the largest IT companies such as IBM support Open Standards and Open Source these standards may not be widely adopted and there can be competing Open Standards in existence.

JORUM seeks to implement Open Standards as much as possible, especially in the areas of learning object metadata, content packaging, digital rights management and so on. In order to comply with the JISC IE the JORUM service repository needs to support:

- The IMS Content Packaging Specification
- For distributed searching, either Z39.50 or SRW/SRU
- The Dublin Core record format defined by the SRW DC XML schema, for SRW (Search and Retrieve Web Services).
- The return of metadata records that conform to the UK LOM Core using the IEEE LOM XML schemas.
- An OAI-PMH interface should be implemented to allow harvesting of metadata

Early results (CETIS codebash, March 2005[4]) seem to show that this is leading to a high level of interoperability between the various e-Learning tools used in the HE community.

One of the benefits of the development and use of Open Standards and Specifications is the attempt to develop software toolkits and Web services that can work together, or interoperate.

Standards apply to programming languages, which are the basic 'tools' for building all the other software 'Tools'. One of the most prevalent computer languages used today is the Java language, developed originally by Sun Microsystems. So far, Sun has been the 'guardian' of Java, as a proprietary standard. The Apache Software Foundation has recently announced that, with Sun's approval, they will develop an open source version of Java under an Apache licence. This will be an implementation of J2SE (Java platform 2 Standard Edition, which does not include implementations of many specifically server-side technologies included in J2EE, the

[1] Available from: http://www.perens.com/OpenStandards/Definition.html
[2] Available from: http://linuxlab.dk/openstandards/#what
[3] Available from: http://www.oss-watch.ac.uk/events/2003-12-11/open-standards.pdf
[4] Third CETIS/LIFE codebash available at: http://www.cetis.ac.uk/content2/20050329191619

Enterprise Edition). Project Harmony was announced in early May 2005 and promises to be a mammoth undertaking.[1]

> A key reason why Sun has expressed reluctance to make Java open source is compatibility--the guarantee that a Java program will run on any Java software foundation. It's not an academic concern: Sun fought for years with Microsoft after the software giant added extensions to Java that broke compatibility. If Java were open source, it also would permit people to create incompatible versions of that software.[2]

The software of several Institutional Repository projects is written in Java (e.g. Fedora, DSpace) and the servlet/JSP container Tomcat runs on Java. Apart from Jakarta Tomcat, many other Apache projects run on Java, such as Apache Lenya, an Open-Source Content Management System, built on top of Apache Cocoon (itself Java based). [3]

5.5 The Toolkit Approach

There is a growing number of OSS/FS tools that can be used to build a LOR, and various LORs described below include these tools. This was the experience of City College Coventry Content Repository (CCCCR) in the UK. The Canadian e-Learning programme have also contributed to the development of such tools like the eduSource Communication Layer (ECL) – see section 5.5.3.

To quote a personal communication from Rob Talbot, designer of CCCCR:

> Building a Learning Object Repository from Open Source components is entirely feasible. However, Open Source provides a toolbox rather than a built solution. Work has to be done to connect up components and provide a suitable interface.

5.5.1 The Apache Cocoon project

This project, allied to the Apache Software Foundation, is described as "web glue for your web application development needs". It is a glue that keeps concerns separate and allows parallel evolution of all aspects of a web application, improving development pace and reducing the chance of conflicts.

Apache Cocoon is an XML publishing and application framework with a particularly vibrant development community with thousands of users and many active core developers. Apache Cocoon uses the component pipelines model as well as the key concept of Separation of Concerns (SoC). SoC assigns the basic functionalities of a completed application to separate teams, allowing them to work on their own functions without being hampered by the other teams. The pipeline passes SAX (Simple API for XML) events from one end of the pipeline to the other. In this way Apache Cocoon separates the basic components, their aggregation and how the resulting functionalities are deployed in the final application. Apache Cocoon has broad standards compliance, supporting many recent XML technologies (e.g. XPath, XQuery and much more) as well as established standards for data source integration (including JDBC, LDAP, Web Services and WebDAV).

In this way (see below) Orixo claim that Apache Cocoon provides a scaleable and coherent architecture that can tie the 300+ existing Cocoon components together to build complex Web applications.

[1] Article available from: http://uk.builder.com/programming/java/0,39026606,39245888,00.htm
[2] Article available from: http://news.zdnet.com/2100-3513_22-5256228.html?tag=nl
[3] Apache Java projects http://jakarta.apache.org/site/java_at_apache.html

5.5.2 Apache Lenya

MIT has examined an OSS/FS Content Management System (CMS) called Lenya[1] that runs under an Apache/Cocoon Web framework, all open source. Lenya was developed by a group in Germany called Wyona. In June 2004, the Lenya CMS was donated by Wyona to the Apache Software Foundation and is currently being incubated as a Cocoon subproject[2] .

Lenya is an Open-Source Content Management System written in Java and based on open standards such as XML and XSLT. Lenya is built on top of Apache Cocoon and other components from the Apache Software Stack. Its XML-centric architecture allows for content delivery targeted to the capabilities of various user agents (e.g. web browsers, PDAs). Apache Lenya is built around off the shelf components from the Apache Software Foundation. Lenya comes with the features you can expect of a modern Content Management System, such as Revision Control, Scheduling, a built-in Search Engine, separate Staging Areas, and Workflow.

Orixo[3], an alliance of Open Source development companies spanning six European countries, was officially launched in June 2003. Orixo members are established developers of OSS/FS with a demonstrable commitment to projects such as Apache Cocoon.

5.5.3 eduSource Communication Layer (ECL)

The ECL tool has been used in a number of repository systems worldwide, in conjunction with other tools, such as the eduSource Repository-In-A-Box (eRIB) from Canada. ECL is used as a component in the Commonwealth of Learning's learning object repository – COLLOR.

The ECL enables interoperation between distributed repositories and was one of the first applications of the IMS Digital Repository Interoperability specification. The ECL is being developed and supported by the LORE Research Group[4] at the Simon Fraser University. LORE is the Laboratory for Ontological Research researching the infrastructure for object repositories.

Since the end of the eduSource project in March 2004 the ECL development continued with support from the LORNET project and LionShare projects as part of the larger research goals of LORE lab.

> ECL Protocol is the language for all eduSource Network members connected to a heterogeneous network. It consists of institution repositories, peer-to-peer network, individual small repositories, and application interfaces. The institution repositories such as Explora, Aloha, and Careo are stable eduSource members. Typically, the institution repositories have known URL and Web servers running at all times. These repositories already offer their services through their own protocol either via HTTP, XML-RPC requests or other protocols. ECL protocol enables these repositories to communicate with each other and enables other repositories and services to become a part of eduSource network. The protocol is independent from any existing protocols and enables developers to build universal tools and services that will enable their users to connect and use services provided by any repository connected to the eduSource network.
>
> The ECL implements the IMS Digital Repository Interoperability specification. It follows closely IMS DRI and models its messaging protocol according to IMS DRI general messaging model.
>
> ECL is an infrastructure for connecting learning object repositories and services into the open network. The current development of the ECL addresses the issue of discoverability of services (from the end-user perspective) and security. The ECL

[1] Apache Lenya http://lenya.apache.org/
[2] The Apache Cocoon project Lenya http://cocoon.apache.org/lenya/
[3] Orixo website at http://www.orixo.com/index.html
[4] LORE Research Group at: http://lore.iat.sfu.ca/

Security is developed within the LionShare project[1] and will provide repository access for authenticated P2P users. [2]

The ECL allows repositories implemented on different platforms and using different metadata schemas to connect into a single network. The ECL defines actions that correspond with IMS DRI main functions: search/expose, submit/store, gather/expose and request/deliver.

5.5.4 ALOHA

The open source ALOHA[3] java client tool was developed out of research early in the CAREO project. ALOHA takes advantage of an open XML-RPC API created to work with CAREO repositories so it can plug into the metadata resources of multiple repositories. Based on the successful RELOAD tool, ALOHA improves the workflow of objects into such repositories allowing indexing of resources using any metadata standard. It has drag and drop functionality that can automatically extract metadata from over 200 file types. The administrative job of marking up metadata can thus be made much easier, especially for diverse groups of users, such as librarians, teachers and media developers.

[1] Lionshare project http://lore.iat.sfu.ca/projects/index.html#LionShare
[2] An ECl Overview available from: http://lore.iat.sfu.ca/ecl/#LORNET
[3] ALOHA http://aloha.netera.ca

6. Overview of International Initiatives and Projects

6.1 Introduction

Internationally, many countries have well-developed e-Learning programmes that include the use of various types of digital repositories. These resource services usually need to exchange data with other parts of the programme or in some cases other parts of the world. Due to the diverse services and resources that typically exist within such e-Learning programmes there is a need to ensure as high a degree of interoperability as possible. If services are further broken down into their constituent components, then the need for common communication protocols and standards becomes obvious.

The existence of a wide range of proprietary software systems that support e-Learning, each catering to its own market, has not always led to a standards-based, interoperable learning environment. Proprietary, closed standards can act to impede such an integrated situation. Certain proprietary software (e.g. WebCT) have a tendency to bundle together learning tools, activities, and resources that are very useful in a single learning environment, but are often opaque to any other tools, developed outside their company. By taking a wider, or more 'global', view of the way resources and tools need to be combined together, then a more integrated approach can, in theory, be developed.

The 'Building Blocks' approach of the Blackboard VLE is a more 'open' system, where the core functionality of the software is retained in the Blackboard software, but added features can be developed by 3rd party developers[1].

As mentioned in section 5, the e-Learning Framework (ELF) in which the JISC participates, takes a particular view of how the various e-Learning tools and services need to fit together. The ELF is an international collaboration, emphasising the fact that the design of e-Learning systems, including learning object repositories, needs to operate in a standards-based international arena to maximise their usefulness to users.

The Web Services specification guides the Service Oriented Architectures (SOA) approach of the ELF.

The UK JISC has invested in some demonstrator projects investigating how "service definitions" and toolkits can be used in practice. The Discovery + (Brokerage for Deep and Distributed e-Learning Resources Discovery) project aims to develop a toolkit corresponding to the resource discovery services area of the JISC e-Learning Service-Oriented Framework.

> The project develops a software development toolkit (SDK) corresponding to 3 common services areas of the proposed services oriented JISC e-Learning Framework, i.e. federated search, search and resolver services. The SDK mediates the discovery of deep resources in distributed and heterogeneous repositories.[2]

One well-known example of a specific tool, widely used by developers, is the JISC-funded RELOAD[3] tool that creates Content Packages which can then be imported into the JORUM repository.

The creation of component tools is paralleled by work outside the UK:

- the Canadian e-Learning programme where federations of digital repositories are being connected up to provide access to a much larger number of learning objects. The eduSource project has also produced various tools for the management and metadata tagging of learning objects.

[1] Tall Desk Study report available from: http://www.elframework.org/projects/tall/bodington/
[2] Project website for D+ http://devil.lib.ed.ac.uk:8080/dplus/
[3] RELOAD http://www.reload.ac.uk

- the Apache Cocoon project adopts a “toolkit” approach with the development of the Content Management System called “Lenya” and other initiatives.

Work on standards and specifications is international and this is particularly true of the CORDRA project. Content Object Repository Discovery and Registration/Resolution (CORDRA) seeks to create interoperable registries of content and content repositories. Whilst other reports[1] consider this in more depth, it seems likely that any future JORUM repository will want to consider implementing the required specifications to enable JORUM to be federated with other, international repositories. This is an area of work that JORUM continues to watch.

6.2 Canada

Canada has perhaps seen the most active interest in using OSS/FS to supply their e-Learning infrastructure in terms of LOR-related software.

The Canadians have invested in some high-profile initiatives e.g. CANARIE Inc. (launched in 1999) e-Learning program which has invested close to $30 million in online learning projects. The design and development of Learning Object Repositories (LOR) constituted a major research theme of the program. There have been LOR-based projects producing valuable research in software, standards, community building and infrastructure deployment e.g. the CanCore application profile, the eduSource Communications Layer (ECL) - for interoperation between distributed repositories.

Significantly for the current report, the Canadian initiatives have made internationally recognised contributions to the creation of open source tools that, added together, bring closer the possibility of a fully open sourced infrastructure of LOR systems. MacLeod describes six tools in the recent CANARIE report on Learning Object Repositories[2]. These tools include the TILE authoring tool, as well as various metatagging, federated search and object management tools.

6.2.1 eRIB

The eduSource project produced the demonstration system known as eRIB - eduSource Repository-In-A-Box – an open source suite of tools for searching, metatagging and managing learning objects. With CANARIE's support, a consortium of more than 35 university, government and industry partners have developed open-source tools, systems, protocols and practices for a new pan-Canadian repository that seamlessly links multimedia databases from across the country via the Internet. Fully bilingual and accessible to all Canadians, including those with disabilities, eduSource makes it easy for users to catalogue, post and share learning objects. Educators can download content from eduSource to build whole curriculums and courses of study. EduSource also links to repositories in the U.S. and Australia.

The eduSource Repository-In-A-Box (eRIB) has seen significant international uptake, especially in developing countries such as India and parts of Africa.

6.2.2 CAREO

CAREO[3] (Campus Alberta Repository of Educational Objects) is a project supported by Alberta Learning and CANARIE (Canadian Network for the Advancement of Research in Industry and Education) that has as its primary goal the creation of a searchable, Web-based collection of multidisciplinary teaching materials for educators across the province and beyond. CAREO takes a distributed approach to the location of “educational objects” in its architecture.

[1] Technical Frameworks and Infrastructures Report (port (due 2/9/05)
[2] ‘Learning Object Repositories: Deployment and Diffusion’ D. MacLeod, February 2005 available from: http://www.canarie.ca/funding/elearning/2005_LOR_report.pdf
[3] CAREO http://www.careo.org/

6.2.3 APOLLO

APOLLO (Academic Publishing for Online Learning with Learning Objects) was initially chosen as a 2nd generation to the CAREO software to provide LOR functionality by BCcampus. BCcampus was established in 2002, with a mandate to provide British Columbia learners with a web-based access point to online learning programs and services – a brokerage service.

6.2.4 COLLOR

COLLOR[1] – the institution known as the Commonwealth of Learning (COL) have created an OSS/FS LOR which integrates two OSS/FS packages, pakXchange and eRIB. pakXchange implements a document repository system, providing comprehensive role-based administration, and version control, of publications; the eRIB provides the repository functionality. The COLLOR demonstration system is configured to access the eduSource network of repositories. The system includes the ability to add information about the learning material ("learning objects") with "meta tags". These are IMS-compatible (IMS Schema 1.2.2) and the system is able to "federate-search" other repositories when the user is looking for learning materials.

Along with the OSS/FS developed by the eduSource and CANARIE projects COL has worked with Vancouver-based 3waynet Inc. to develop some complementary OSS/FS that, when used together with the project software, could provide a fully functioning learning object repository. In co-operation with COL, 3waynet created all the programming "connective tissue" to make these systems work together and provide an easy-to-use webpage to access content. COL is hosting the software and is collaborating with the African Virtual University, headquartered in Nairobi, Kenya, that will upload and make available open source courseware that Commonwealth countries can access free of charge. COL anticipates partnering with other groups as awareness of the LOR grows.

6.2.5 TILE

TILE [2]– "The Inclusive Learning Exchange is a revolutionary learning object repository service that responds to the individual needs of the learner. TILE provides the authoring tools, repository architecture, and preference schema needed to support this learner-centric transformation, particularly to allow learning objects to be accessible to those with disabilities … TILE is now evolving to become a service layer that can sit on top of learning object repositories and allow users to search and access them in a responsive way".

CAREO is probably the best-known of these systems to date. The repository contains just over 4000 objects at the moment, which are loosely defined as modularised educational resources, educational objects or learning objects. The system is designed to be part of a larger networked repository system that displays XML document records based on IMS metadata. CAREO uses the CanCore Application Profile for metadata tagging of objects.
The CAREO application is designed to allow the search, retrieval and display of IMS metadata records in a web browser. Apart from tools supporting resource discovery (e.g. search, retrieve) it enables publication, aggregation and sharing of online materials. CAREO implements IMS Content Packaging to allow users to aggregate content.

> These compound aggregations may range from simple collections of images into a single package, to electronic representations of physical books, to highly structured online courses.[3]

The University of Calgary Learning Commons[4] repository uses CAREO as do several other repository systems.

[1] Commonwealth of Learning http://www.col.org/lor/
[2] TILE http://www.inclusivelearning.ca/tile/index.html
[3] Building Digital Books with Dublin Core and IMS Content Packaging, M. Magee et al, Netera alliance
[4] Learning Commons http://commons.ucalgary.ca/about/index.html

6.3 USA

In some sectors of the USA there is a substantial amount of faith in the OSS/FS method; in particular, the use of secure Linux servers in the Department of Defence. There is immense diversity in this large market and many educational institutions are looking closely at developing OSS/FS solutions.

There are several systems based on DSpace today that are described as LORs, though none seem to use IMS Content Packaging or be SCORM compliant.[1]

6.3.1 DLearn

DLearn[2] is a learning object repository of shareable, digital, learning materials developed at the University of Arizona that is based on the DSpace model. The University of Arizona has re-tooled DSpace as a LOR to become "DLearn." They retained Dublin Core (DC) metadata in favour of LOM.
Like other repositories following the DSpace model, DLearn content is organized around communities which correspond to University of Arizona administrative entities such as schools, departments, labs and research centres. Within each community there can be an unlimited number of collections. Within each collection there can be an unlimited number of items. Content will consist of lectures, journal articles, book chapters, syllabus, music recordings and data sets in various digital formats.

6.3.2 MatDL

MatDL[3] is the Materials Digital Library, and is part of the National Science Foundation (NSF) MatDL is a collaborative effort of the Materials Science and Engineering Laboratory at the National Institute of Standards and Technology (MSEL/NIST), Kent State University, Massachusetts Institute of Technology (MIT), University of Michigan, and the University of Colorado at Boulder.

6.3.3 The ELATED project

The following describes the role of the ELATED application.

> ELATED is an application that was designed to operate on top of the Fedora repository system. ELATED provides a simple user interface and set of functionality that allows Fedora to be used as the back-end to a general-purpose digital assets management system. The goal of this software is to provide a usable program that small institutions may use to manage growing collections of digital data, and to do so in a manner that is consistent with the Fedora project so that these same institutions may adopt or transition to other tools that will be developed in the future.
>
> ELATED allows all users to browse collections with public visibility. Once logged in, users have the ability to create and manage their own public or private collections. ELATED also allows for shared--or collaborative--collections to which multiple users my contribute. Collaborative collections contain a simple workflow system that allows editors to monitor and approve of submissions.[4]

6.3.4 The Visual Understanding Environment

> The Visual Understanding Environment (VUE) is an information management application that provides an interactive, concept mapping interface to digital

[1] Personal communication from MacKenzie Smith, MIT Libraries
[2] DLearn at http://www.dlearn.arizona.edu/index.jsp
[3] MatDL at http://www.matdl.org/
[4] From the ELATED project website http://spumoni.colleges.org/elated/publicInterface.jsp

resources accessed via the web, from FEDORA-based digital repositories, ftp servers or local file systems.

The Visual Understanding Environment (VUE) project at Tufts' Academic Technology department provides faculty and students with flexible tools to successfully integrate digital resources into their teaching and learning. VUE provides a visual environment for structuring, presenting, and sharing digital information and an OKI-compliant software bridge for connecting to FEDORA-based digital repositories. Using VUE's concept mapping interface, faculty and students design customized semantic networks of digital resources drawing from digital libraries, local files and the Web. The resulting content maps can then be viewed and exchanged online. This project is supported by the Andrew W. Mellon Foundation.[1]

6.3.5 Spoken Word Project

As part of the international JISC/NSF Digital Libraries in the Classroom Programme led by Glasgow Caledonian University, Northwestern University and Michigan State University in partnership with the BBC this project explores the use of digital audio in the humanities.

> The Spoken Word proposes to transform undergraduate learning and teaching through integrating the rich media resources of digital audio repositories into undergraduate courses …[2]

Fedora is one of the repositories used and was chosen for several reasons, including its support for preservation activities such as content versioning and its ability to meet their pedagogical objectives. The OSS/FS nature of the product, plus its interoperability, rights awareness, scalability and ability to allow content repurposing were the main attractions of using Fedora for the project.

6.3.6 Connexions

Connexions[3] at Rice University is an environment for collaboratively developing, freely sharing, and rapidly publishing scholarly content on the Web. This system is based on the Plone[4] Content Management System, which runs on Zope[5], an open source application server. The system's collections are made up of modules, each an XML document meeting specific criteria allowing their use and reuse in various contexts. Each item is written in cnxML, a format that contains both the metadata for a material and the content itself. Connexions is focused on the web-publishing of shareable modules (instead of learning objects per se) with the aim of building up non-linear courses out of these modules.

> The Connexions project at Rice University is a collaborative, community-driven approach to authoring, teaching, and learning. By collaborating both within and across disciplines, communities of authors work together to pool their expertise in the form of knowledge modules. These modules form the basis for building courses that are authored by many, with each author receiving attribution for his or her contributions. Information can be modified under an open licence to tailor the material for the audiences of learners.[6]

[1] From the VUE website http://vue.tccs.tufts.edu/about/
[2] Spoken Word http://www.spokenword.ac.uk/
[3] Connexions http://cnx.rice.edu/aboutus/
[4] Plone http://plone.org/
[5] Zope http://www.zope.org/
[6] Henry, G. 'Connexions: An Alternative Approach to Publishing' ECDL 2004 European Conference on Digital Library, University of Bath, United Kingdom, September 2004 available from: http://www.ecdl2004.org/

The philosophy[1] of Connexions involves:

> A global community of authors continuously converts "raw knowledge" from the continuum into small, self-contained *modules* of information, the equivalent of a page or two in a textbook. Modules can be imagined as special Web pages with hyperlinks pointing to prerequisites, applications, and supplementary material. Modules are placed in the *Content Commons* (a database repository) to be used, reused, updated, and adapted. Instructors use a *Course Composer* software tool to weave modules into *customized courses* that can be placed on the Web, presented in class, or printed as a paper text. Students and other learners access Web courses or the Content Commons directly, using special *visualization and navigational tools* designed to highlight the non-linear "Connexions" among concepts both within the same course and across courses and disciplines.

6.4 Europe

There are several collaborative e-Learning projects using digital repositories undertaken by the European nations.

6.4.1 ARIADNE

One of the best known is from the ARIADNE Foundation which aims to foster the share and reuse of Learning Objects, both by universities and corporations. ARIADNE began the work on Learning Object Metadata (LOM) which was later adopted by the IMS Global Alliance to form the IMS LOM specification.

The ARIADNE Knowledge Pool System is a European distributed learning object repository for multicultural/multilingual teachers and learners.

ARIADNE tools are:

- Ariadne Web Services (AWS) offer a simple way of accessing the ARIADNE Knowledge Pool System for programmers
- SILO (Search & Index Learning Objects) is a tool for end-users to search in the ARIADNE Knowledge Pool System
- WebLe (Web-Based Learning Environment) is a Learning Management System, linked to the Knowledge Pool System

ARIADNE is considering using OSS/FS but has not done so as yet.

European projects offering advice and guidance about using OSS in Education include:

6.4.2 SIGOSSEE

The Special Interest Group in Open Source Software for Education in Europe (SIGOSSEE) has been established to investigate, inform, and advise the education community on the uses and benefits of Open Source Software and Open Content. Some of the activities of this SIG include the development of a catalogue of examples of good practice. A Website has been set up to publicise these activities[2].

6.4.3 PLANET

PLANET Digital Repository[3] is an e-Learning project led by the Universitat Rovira i Virgili in Tarragona; their digital repository implements the open source ECL protocol (based on IMS DRI) using web services. This provides search submit and request services, with search being

[1] Connexions White Paper, Revised August 2004 available from: http://cnx.rice.edu/aboutus/publications/ConnexionsWhitePaper.pdf
[2] http://www.ossite.org
[3] http://ants.etse.urv.es/planetdr/

based on LOM metadata. PLANET Digital Repository is Java-based, runs on Jakarta-Tomcat and uses the open source MySQL database.

6.4.4 Railroad Repository

The Infrae 'Railroad Repository'[1] for Zope/Plone.

Railroad is a standards based repository for binary files such as digital media, along with their metadata. It is designed to be easy to integrate with content management systems and other client software.

There seems to be some important OSS/FS developments in the UK, apart from the awareness-raising function of CETIS in terms of communicating the latest developments about standards for interoperability and the investment from JISC in terms of developing an e-Learning Framework using Web services. Much of the work so far has taken place in the field of Institutional Repository systems which include the following two projects.

6.4.5 Cambridge University

In the UK, Cambridge University[2] is using DSpace as a digital repository; as a system to store digital information and its descriptive "metadata". Its main purpose at Cambridge is to capture and preserve academic and related content and to make it available online. Although the full IMS Learning Object Metadata is not used in this instance of DSpace, Cambridge does extend the metadata fields to match its requirements. DSpace typically organises materials within collections known as “Communities”. The Cambridge, UK, instance of DSpace has Communities for Archaeology, Music, Social Anthropology, and others.

Cambridge’s DSpace instance is optimised for the Library community and serves as a complete application where the goals are dissemination of outputs and preservation.

Elsewhere In Europe, DSpace is used by The Open University of the Netherlands[3] for their digital repository of Research & Development into learning technologies.

6.4.6 EPrints

As stated previously, EPrints, developed by the University of Southampton, UK, has the largest user base for self-archival repository software found in the recent Open Society Initiative Guide to Institutional Repository software[4]. The EPrints website[5] lists 155 archives[6] using GNU EPrints, used internationally including the US, Canada, Europe and Australia.
OpCit, the Open Citation Project[7], uses GNU EPrints software alongside other tools. The University of Southampton uses GNU EPrints to make its growing database of research literature available to the world through e-Prints Soton[8].

6.4.7 Coventry

City College Coventry Content Repository is an example of an Institutional Repository (IR) whose initial target audience was primarily the staff of City College Coventry who needed more convenient access to National Learning Network learning objects produced within the UK. It is a very small-scale venture, though several other colleges have installed copies of this software for similar purposes.

It uses entirely OSS/FS components in its construction, such as Perl modules (CPAN) and XML derived from the Gnome project (LibXML2, LibXSLT).

---

[1] http://www.infrae.com/products/railroad
[2] http://www.dspace.cam.ac.uk/
[3] The Open Universiteit Nederland at http://dspace.learningnetworks.org/index.jsp
[4] OSI Guide http://www.soros.org/openaccess/pdf/OSI_Guide_to_IR_Software_v3.pdf
[5] Archives using GNU EPrints http://software.eprints.org/archives.php
[6] As of 28/07/05
[7] OpCit http://opcit.eprints.org/
[8] University of Southampton research literature database http://eprints.soton.ac.uk/

6.4.8 Luminas

As the Orixo (see section 4.5.1) presence in the UK, Luminas[1] has a track record of technology provision for JISC projects in the 5/99, X4L (Arches project at University of Warwick[2]) and other programmes where it has built web applications which make use of Apache Cocoon in delivering repository architectures for community-based cultural heritage projects. It has worked with the BBC using Apache Cocoon. Luminas re-wrote the Bristol BioMed repository to use Apache Cocoon, building a custom component to allow collections of archival objects to be made OAI PMH aware. This will make the collections harvestable by OAI harvester engines.

The BioMed Image Archive[3] is an online collection of around 8000 medical, dental, and veterinary images for use in learning, teaching and research. All the images have been donated by academics working in the biomedical fields in different countries.

As discussed in section 4.5, Luminas use the 'toolkit' approach involving component-based application development and the "plug and play" programming paradigm.

6.5 Australia

In the Australian e-Learning context, Education Network Australia (EdNA Online[4]) is one their largest education and training resources and since 1996 it has been providing the education and training community with online access to quality assured information and tools and services to support teaching, learning and research.

The new generation EdNA Online search centres provide access to other quality assured repositories of education and training resource information: both Australian and international. A visitor to EdNA Online can search just one repository, or they can mix and match repositories.

6.5.1 BELTS

BELTS (Basic E-learning Tool Set)[5] has been developed by The Le@rning Federation (TLF)[6] to demonstrate the distribution, management and use of online curriculum content and to aid investigation of requirements for e-learning environments by Australian and New Zealand school jurisdictions. An overview of BELTS and full installation details can be found at the SourceForge project website[7]. BELTS is an OSS/FS product.

BELTS currently provides a limited set of tools, including:

- A content repository
- Basic activity creation, using lessons
- Basic group management, using classes
- Content to curriculum outcomes matching (the curriculum organiser)
- Downloading of content from The Le@rning Federation's Exchange repository of online curriculum content
- Content replication from one BELTS to another, and
- System administration

BELTS has currently not been developed as a fully featured learning management system. but is, however, an OSS/FS project that can be further developed. The Le@rning Federation encouraged Australian and New Zealand education jurisdictions, and others, to consider options for collaborating and contributing to the evolution of BELTS.

---

[1] Luminas Ltd. website http://www.luminas.co.uk/index.html
[2] ARCHES http://www.warwick.ac.uk/ETS/arches/
[3] The Bristol BioMed Image Archive http://www.brisbio.ac.uk/
[4] EdNA Online http://www.edna.edu.au/edna/page1.html
[5] BELTS http://www.thelearningfederation.edu.au/tlf2/showMe.asp?nodeID=98
[6] The Le@rning Federation (TLF) http://www.thelearningfederation.edu.au/
[7] http://belts.sourceforge.net/

Education Queensland used the BELTS as part of its digital resource collection management within the Learning Place[1]. The Learning Place is Education Queensland's integrated online learning platform, supporting online learning, resource discovery and communication.

"BELTS was developed as part of The Learning Federation project. However, we wanted a more sophisticated system"[2], said Steve Schreiber, Project Director, Online Learning Infrastructure for Education Queensland.

A proprietary solution, The Learning Edge, offered by a Tasmanian company, Dytech Solutions product has since replaced BELTS for the Learning Place repository. The Learning Edge product is an advanced J2EE content authoring tool and Learning Content Management System (LCMS), acting as a digital repository. This platform-independent program integrates with leading Learning Management Systems and can be used for a number of functions including managing digital rights, including copyright and moral rights issues, and workflow. It is also SCORM and IMS compliant.

6.5.2 Arrow

According to the Project Manager G.J. Payne[3], ARROW was funded in October 2003 for three years to establish institutional repositories in the ARROW Consortium by the Australian Commonwealth Department of Education, Science and Training (DEST), under the Research Information Infrastructure Framework for Australian Higher Education. The ARROW Consortium comprises Monash University (lead institution), National Library of Australia, the University of New South Wales, and Swinburne University of Technology.

ARROW (Australian Research Repositories Online to the World) exposes the research outputs of the university partners to a wider audience and encourages further collaboration between researchers spreading the benefits of this research more widely. ARROW focuses on institutional digital repositories comprising e-prints, digital theses and electronic publishing. The project will identify and test software or solutions to support best practice in this area.

ARROW was funded on the basis that it would explore the utility of OSS/FS to meet the project's objectives, either alone or in combination with proprietary software. It was also an undertaking of the project that any software developed using ARROW funds would be made available as open source.

The project used the previously referred to Open Society Guide for a comparative study of OSS/FS repository software[4], where Fedora and DSpace were judged to have sufficient features and installed user base and resources behind them for serious consideration. The project required the chosen system to support the OAI-PMH protocol for metadata harvesting. Fedora was eventually selected on the basis of its ability to handle large numbers of objects. Since the OSS/FS field is continually developing, the project will review its software solution annually and in October 2004 it was decided to continue using Fedora.

In April 2004 ARROW established that Northwestern University had implemented a Fedora repository containing over 200,000 items, and the software has since been tuned and tested by the Fedora development team with up to 700,000 objects.

Fedora is not a complete repository solution "out of the box". Fedora provides a repository storage layer, which in the repository context is analogous to what the Oracle relational database software provides in a database application. To build a database using the Oracle functionality one needs applications software, and so it is with Fedora when building a repository.

---

[1] http://www.learningplace.com.au/default.asp
[2] Press release 'Tasmanian IT Firm Wins Major State eLearning Contract' http://www.thelearningedge.com.au/PRelease.pdf
[3] ARROW project http://www.arrow.edu.au/docs/files/Educause%202005%20paper%20-%20Payne%206%20Dec%2004.pdf
[4] OSI Guide to Institutional Repository Software 3rd August 2004 Edn. available at: http://www.soros.org/openaccess/pdf/OSI_Guide_to_IR_Software_v3.pdf

6.5.3 Institutional Repositories (IR)

ARROW sees an IR as being capable of holding any mix of anything that cab be represented digitally. Although various types of eprints (e.g. research papers, theses, archival documents) are the initial priority, management of digital audio, moving images, and multimedia objects comprising a mixture of any of the above formats are in scope but of lower priority.

> The project is aware of interest in our managing learning objects and research data sets and is endeavouring to track these fields to ensure decisions we make now do not preclude adding such objects in the future.[1]

Searching of the ARROW repositories is now possible, showing the various research papers added to the system.

6.5.4 Collaboration with Proprietary Vendors

VTLS[2] Inc. (Visionary Technology in Library Solutions) is a leading global company that announced in January 2004 an image management product built over the Fedora repository software. ARROW approached VTLS to consider further developing the VITAL software to meet the project's requirements and by June a formal partnership between ARROW and VTLS had been established.

> VTLS's commitment to maintaining Fedora offers the prospect of vendor support for the ARROW software beyond the ARROW funding, and the capability to extend the ARROW repository solution to other Australian universities assisted by the vendor's existing customer support facilities.
>
> VITAL is being developed to support workflow management for ingest and maintenance of repository content, and to support access to repository content.[3]

6.5.5 ARROW Partnerships

ARROW is one of the four FRODO (Federated Repository of Digital Objects) projects funded by DEST that share some interdependencies, working together cooperatively on areas of common concern.

- ARROW is looking to the Metadata Access Management System (MAMS) project for access control to repository content through development of a profile of the eXtensible Access Control Markup Language (XACML) to specify access conditions for objects in ARROW repositories.
- The Australian Partnership for Sustainable Repositories project and ARROW are exploring interoperability through consistent application of metadata for similar objects.
- The Australian Digital Theses project is harvesting metadata on Australian theses as the basis of a single search environment for discovery purposes. This service will provide links to theses held online at participating institutions, including those held in ARROW repositories. The ADT project has specified a metadata set which ARROW will support for electronic theses.

6.6 Conclusions

From this review of international initiatives it is clear that considerable development work is taking place regarding generic Institutional Repositories, with growing interest in the creation of LOR.

---

[1] Presentation, February 2005, available from: http://www.arrow.edu.au/docs/files/Info%20Online%20paper-Payne-ARROW%2019%20Jan%202004.pdf
[2] VTLS website www.vtls.com
[3] Op cit ARROW project

The work in Canada shows considerable commitment to investing in the LOR field and in the development of a comprehensive e-Learning framework to support their e-Learning programme. The toolkit approach, using eRIB, has been applied both within and outside Canada, as has the Commonwealth of Learning LOR. Their experience in the use of CanCore metadata profiles and the development of the EduSource Communication Layer (ECL) protocol is as valuable as their development of particular OSS/FS LOR.

Developments in the US such as DSpace and Fedora have attracted considerable interest in the repository world, given the sophistication of the product and the perceived degree of funding and support made available by the Mellon Foundation. The use of DSpace by MIT to host their OCW materials indicates their interest in supporting learning. Fedora software is now being used in a growing number of projects, both in the US and elsewhere. The ARROW project (Australia) has chosen Fedora as the repository layer of their partner institutions. The JISC Digital Libraries programme (e.g. The Spoken Word) preferred Fedora for its interoperability, rights awareness and other features.

European and Australian work has been briefly reported and there is evidence that this work draws on international developments (e.g. using DSpace, EPrints and considerable commitment to using a Java-based toolkit approach - Luminas). Continental European work has not been extensively reviewed, but, considering the numerous references to websites supporting OSS/FS use in various branches of Government (e.g. Spain and Germany) there is clearly great interest in this area. The work of ARIADNE in the Learning Object Metadata area is internationally recognised.

7. Review of OSS/FS Learning Object Repositories

7.1 Introduction

This section examines a number of OSS/FS learning object repositories which provide a closer match for JORUM-stated requirements than the more generic repositories detailed earlier in section 4.2.

7.2 CAREO

CAREO is the Campus Alberta Repository of Educational Objects (http://www.careo.org/)

7.2.1 Background

CAREO is supported by Alberta Learning and Canadian Network for the Advancement of Research in Industry and Education (CANARIE)[1]. The purpose of the project is to produce a web-based, searchable repository of teaching resources.

> Campus Alberta's vision is to enable Albertans to "take courses from any college or university in the province, either on-site, online from their homes, or on the job." CAREO's contribution will be to provide educators with the digital teaching content to make this flexible learning a reality. We will accomplish this not only by providing a location to share and find resources, but also by fostering an online community of educators whose digital materials, expertise, and experience will be exchanged. Although the collected teaching materials will be available to all, those who register as members in this community can contribute their own materials, review existing resources, and contact other members with similar interests.[2]

7.2.2 Description

The following text is taken from the CAREO 'Overview and Goals' document[3] and summarises the principles on which CAREO has been developed

CAREO has developed through four closely-related principles:

1. The reuse of modularised educational resources or learning objects
2. The organisation of these resources through standardised metadata
3. The provision of single-click access to these resources through a distributed repository architecture
4. The continued development and enhancement of these resources through quality control, peer review, reward and support practices

Together, these four principles provide some radically new solutions to a number of problems, some of them quite old, the area of course development and delivery:

- The high cost and low returns involved in the development of large, inflexible courses or courseware

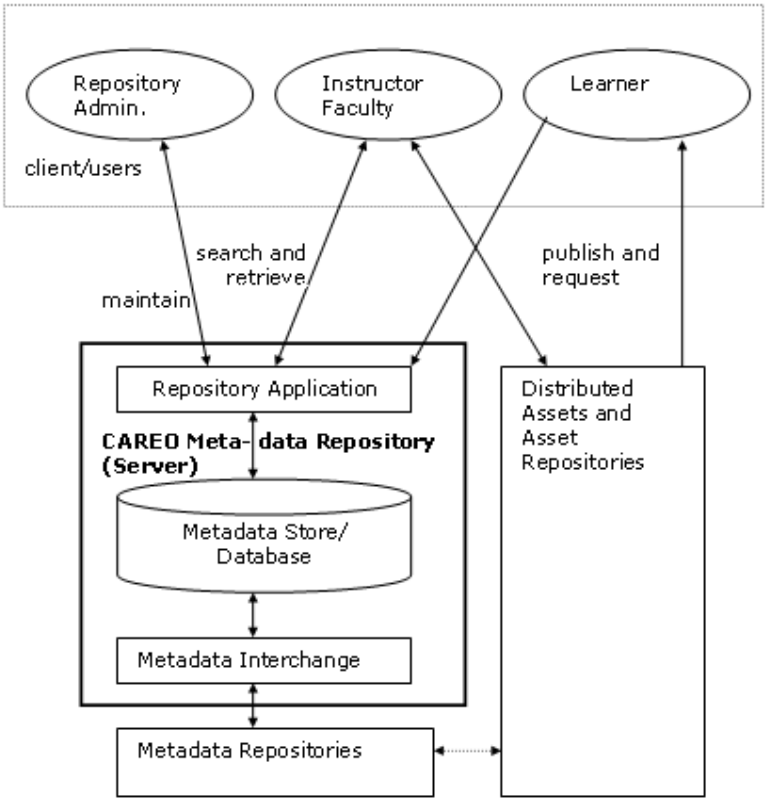
[1] www.canarie.ca
[2] http://www.careo.org/documents/overview.html
[3] http://www.careo.org/documents/overview.html

- Professional recognition for teaching staff engaged in the development or delivery of educational resources
- The maximisation of investments in staff and course development
- The achievement of effective quality control for learning content used in teaching

The CAREO Architecture

CAREO is constructed around a three-tier model, illustrated in figure 7.2.2a

Figure 7.2.2a – CAREO Architecture[1]



The CAREO repository does not store the teaching materials on the CAREO server. All objects are kept at the user's location and the metadata describing the resource is stored within the CAREO system. This is very similar to the MERLOT approach (www.merlot.com), whereby the system only stores a 'library card' record about the resource and its location. However the CAREO repository is somewhat more sophisticated in its functionality compared to MERLOT and other 'Virtual Object Catalogues'.

The repository application provides all the search and retrieval functionality for the user, in addition to all the administrative functionality required to manage a LOR.

The metadata store is the core database for the CAREO repository and is where all the information about the teaching resources is kept. All metadata records use the CanCore Metadata Protocol, which is based on the IMS LOM similar to the application profile used by JORUM.

[1] http://www.careo.org/documents/architecture.html

The CAREO repository is able to link with external LORs or metadata stores like MERLOT. The link with other repositories is done through harvesting techniques and accepted protocols. The harvesting of other metadata and exposure of CAREO's own metadata records is the job of the Metadata Interchange module illustrated in the figure 7.2.2a.

Screenshots

Figure 7.2.2b – CAREO Interface

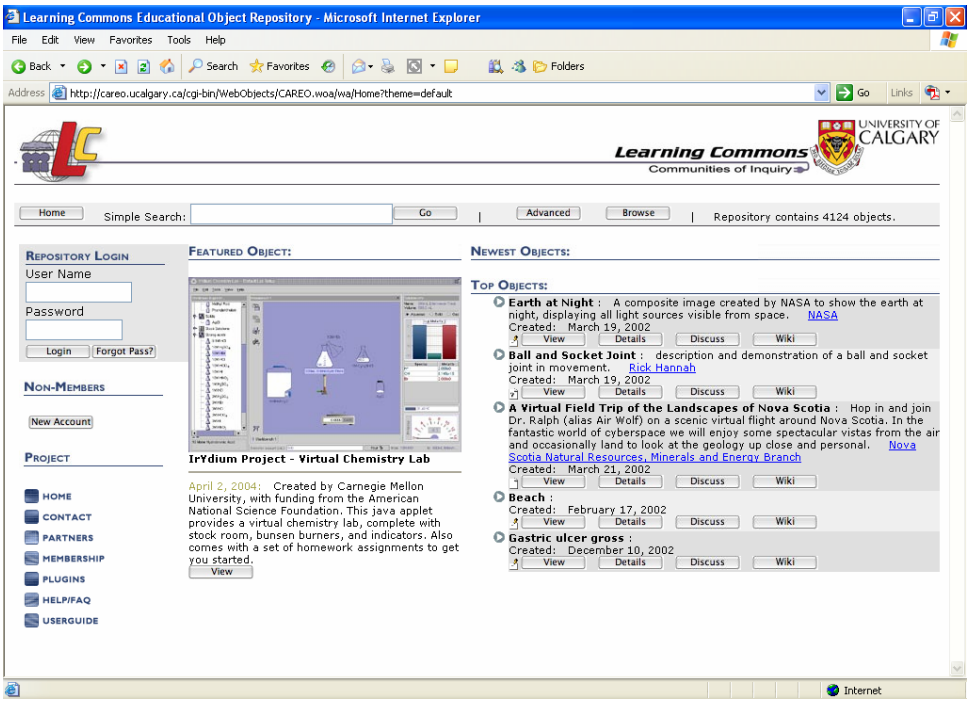


Figure 7.2.2c – Previewing an Object

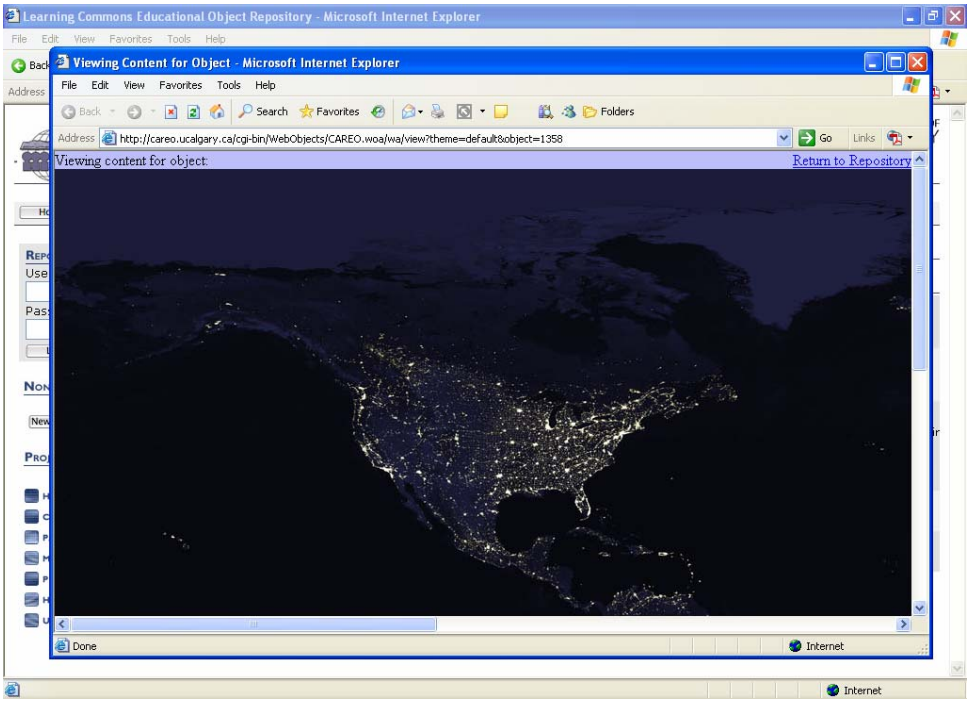


Figure 7.2.2d – CAREO Metadata

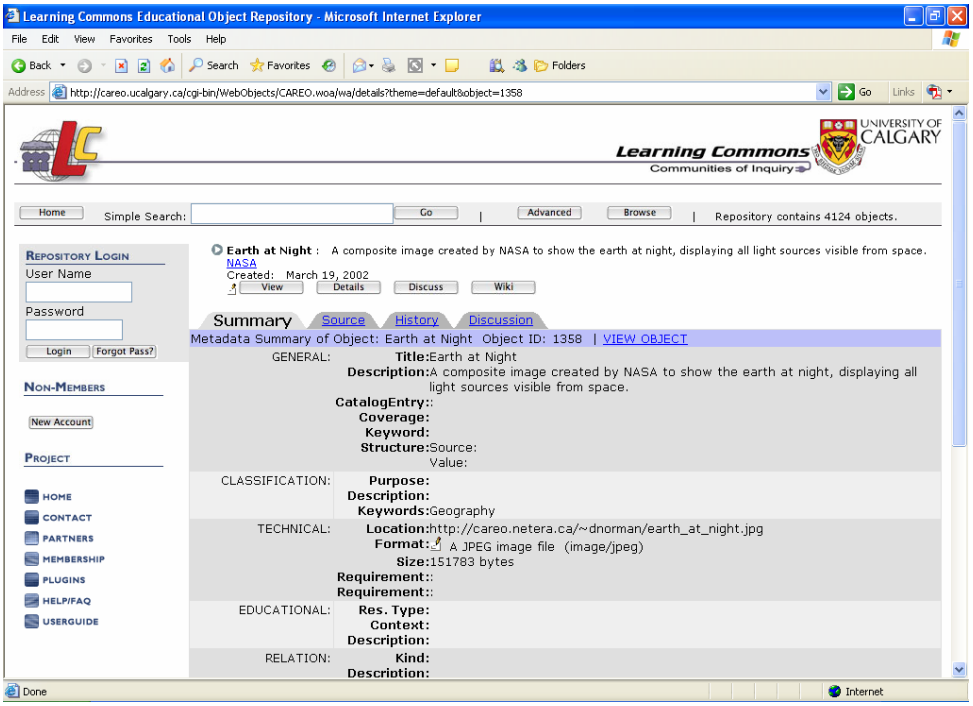


Figure 7.2.2d – CAREO Advanced Search

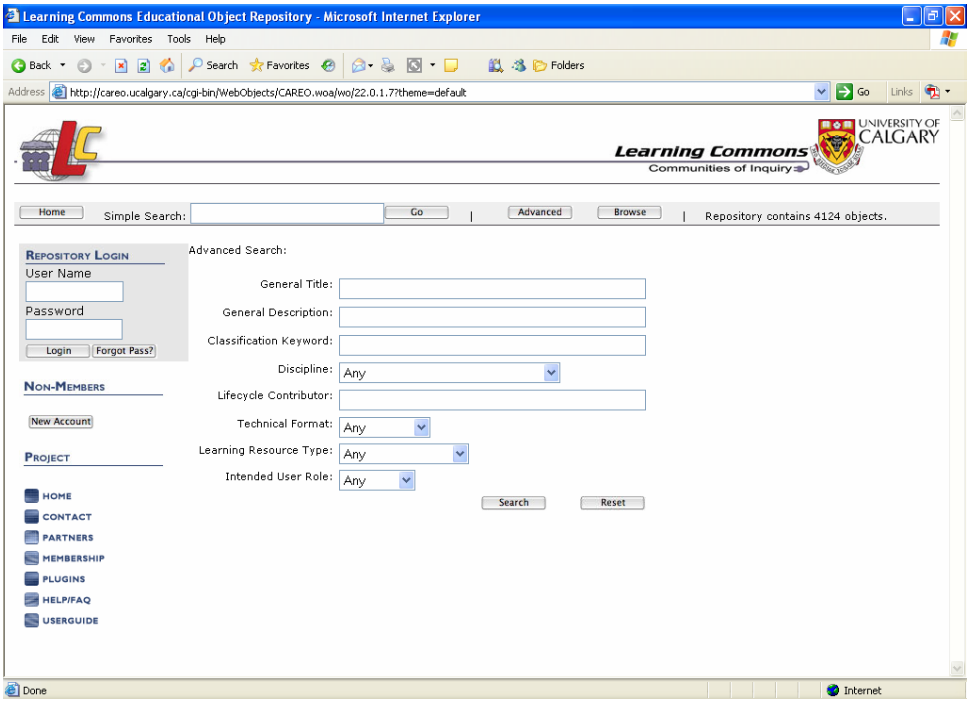
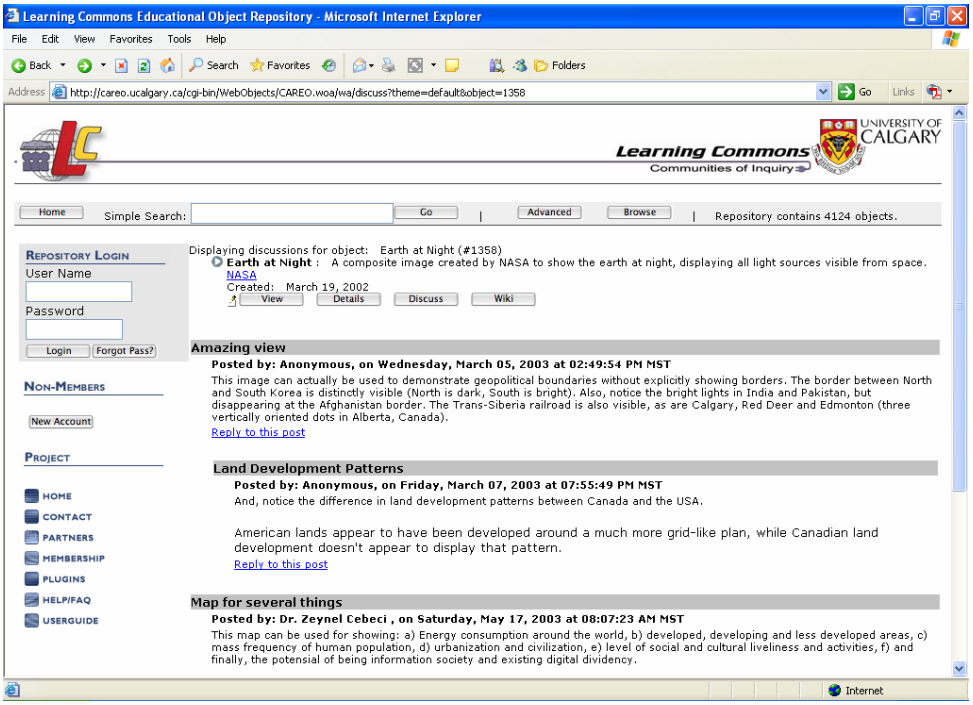


Figure 7.2.2d – CAREO Discussion Forum



7.2.3 Existing Implementations

CAREO has been implemented by the following Universities in co-operation with Broadband Enabled Lifelong Learning Environment (BELLE) and as a part of the Campus Alberta initiative.

1. University of Alberta (http://careo.netera.ca)
2. University of Calgary (http://careo.ucalgary.ca)
3. University of Athabasca

7.3 ALOHA II

ALOHA II is the Advanced Learning Object Hub Application (http://aloha2.netera.ca/index.php)

7.3.1 Background

ALOHA (http://aloha.netera.ca) is supported by the Learning Commons initiative at the University of Calgary.

> Aloha was a project inspired by the diverse needs of educational professionals to index their work in an easily shared and accessible manner. To this end, ALOHA has been designed to accommodate any type of markup, and to allow searching and sharing of those objects. For example, objects marked up in the popular formats IMS and Dublin-Core are both easily handled by ALOHA, unlike other systems which restrict you to a particular format. It's easy to extend a schema to suit your own needs, and yet still have it co-exist alongside other formats. Your document is still searchable along with the rest of the objects. ALOHA ties in seamlessly with online servers, making your work easily accessible by others. Or, if desired, you can restrict your work to a certain group of people, or even to just one person. In addition, the

CAREO website serves as a front end to ALOHA. Objects marked up in ALOHA are immediately accessible through CAREO.[1]

7.3.2 Description

ALOHA II is the next generation of the ALOHA tool undertaken by Netera and the University of Calgary's Learning Commons in partnership with the Bolton Institute in the UK.

> The software has been designed for indexing, aggregating, sharing, multi-purposing, and re-purposing learning objects. It is created to meet the needs of indexers, educators and learners and includes versatile and powerful indexing tools and flexible searching of multiple educational object repositories. ALOHA II incorporates many of the features from ALOHA, but also includes publication to ECL-compliant repositories; customizable forms, controlled vocabularies, and IMS content packaging.[2]

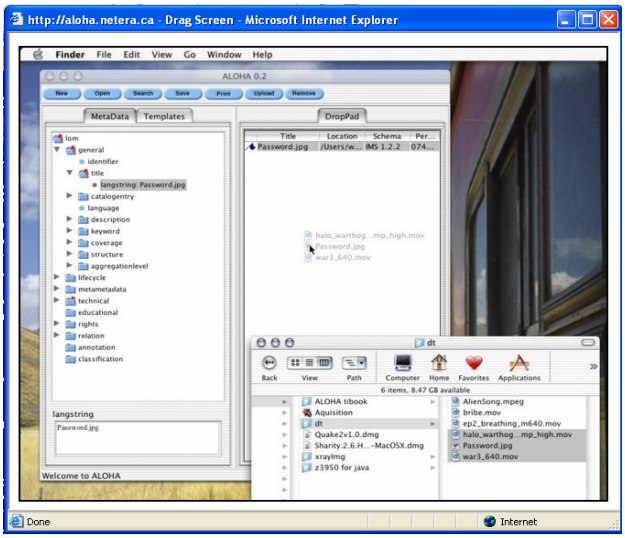
ALOHA II provides the following functionality:

- User-friendly forms to allow learning and teaching resources to be tagged with metadata and then uploaded to an ECL-compliant repository.
- Supports IMS Learning Object Metadata very similar to the JORUM application profile based on the UK LOM.
- Allows the creation, import and export of IMS Content Packages
- Platform Independent
- Custom vocabularies
- Switch between form view and raw XML view
- Spell checker for metadata entry
- Able to search, retrieve and publish metadata records and content packages to existing learning object repositories
- Basic metadata workflow

Screen Shots

All screen shots are of the ALOHA I tool as ALOHA II is still in the final development stages.

Figure 7.3.2a – ALOHA Metadata Tagging



[1] http://aloha.netera.ca/about.php
[2] http://aloha2.netera.ca/project/exec_summ.php

Figure 7.3.2b – ALOHA Preferences

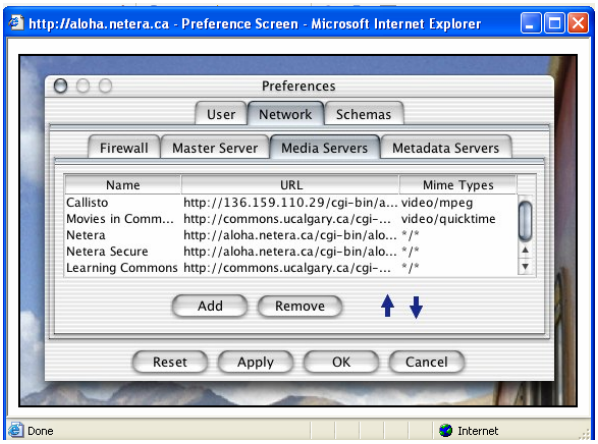
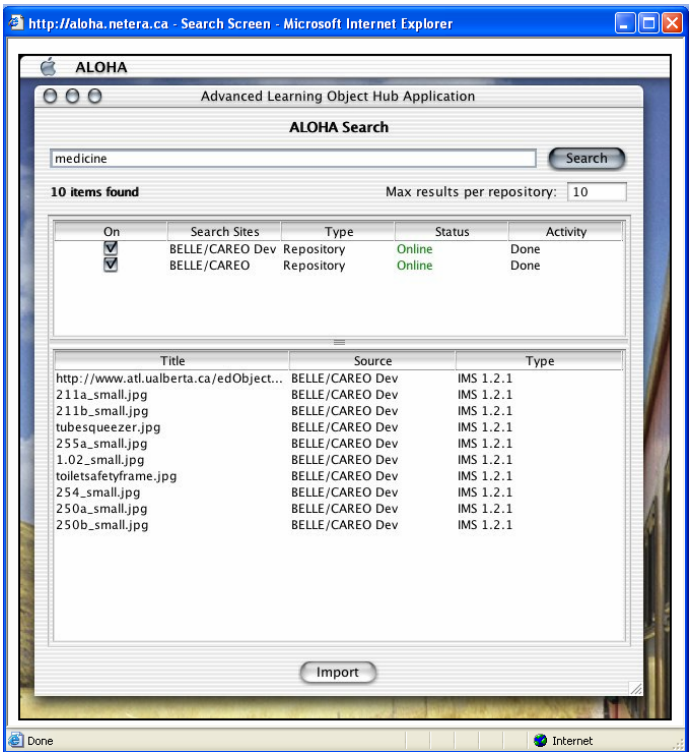


Figure 7.3.2b – ALOHA Search



7.3.3 Existing Implementations

At present there are no projects that have implemented ALOHA as it is in the beta development stage, although a prototype version is available at the ALOHA II website.

7.4 POOL

POOL is the Portal for Online Objects in Learning project (www.edusplash.net)

7.4.1 Background

> The Portal for Online Objects in Learning *(*POOL*)* Project is a consortium of several educational, private and public, sector organizations to develop an infrastructure for learning object repositories. We address the issues of building such architectures including the metadata, software and hardware considerations and bootstrapping the system with initial content. We also make our tools available for download, to help set up similar infrastructures elsewhere and to connect them to *POOL*. The main advantage of our solution is that it can potentially embrace all nature of individuals and organizations involved in the learning object economy[1]

7.4.2 Description

The metaphor used to describe the project architecture is POOL, POND and SPLASH.

SPLASH is the term used for the learning object repository and metadata tagging tool created by the project. SPLASH is different to other learning object repositories as its uses Peer-to-Peer technology instead of a adopting a centralised approach. Each installation of the SPLASH application behaves as a separate LOR allowing the user to administer their own instance of the repository. However the major difference with the SPLASH application is that each installation acts as a node within the whole POOL architecture, allowing all users to search all other SPLASH installations.

> Centralized repositories keep all of their objects in one site, so you have to go through the web portal for that site to search their collection or access their objects. SPLASH uses "peer to peer" technology so every user can have their own mini-repository on their computer, but these are all linked together so you can search all the POOL sites from your SPLASH application.[2]

POND is a metaphor used by the project to describe a community of users. This may simply be a larger, community installation of SPLASH such as an institution, or it could involve building a customised interface to the POOL architecture built on top of POOL Protocols for a community of users.

POOL Central is a specialised peer connected to the network and a high speed Internet. The purpose of the POOL Central is to replicate the queries through the other POOL central peers over the broadband connection and to enhance the reach of the network.

[1] http://www.edusplash.net
[2] http://www.edusplash.net/default.asp?page=FAQ

The table below illustrates the main functionality associated with each component of the project.[1]

	SPLASH	POND	POOL Central
Create/edit metadata record	▲	○	
View metadata records	▲	○	
Search for metadata records locally	●	▲	○
Search for metadata in the POOL network	●	●	▲
Respond to the search request from another peer	●	▲	○
Propagate search query and return collected results	●	○	▲
Robust database support	○	▲	
Management and workflow tools		▲	

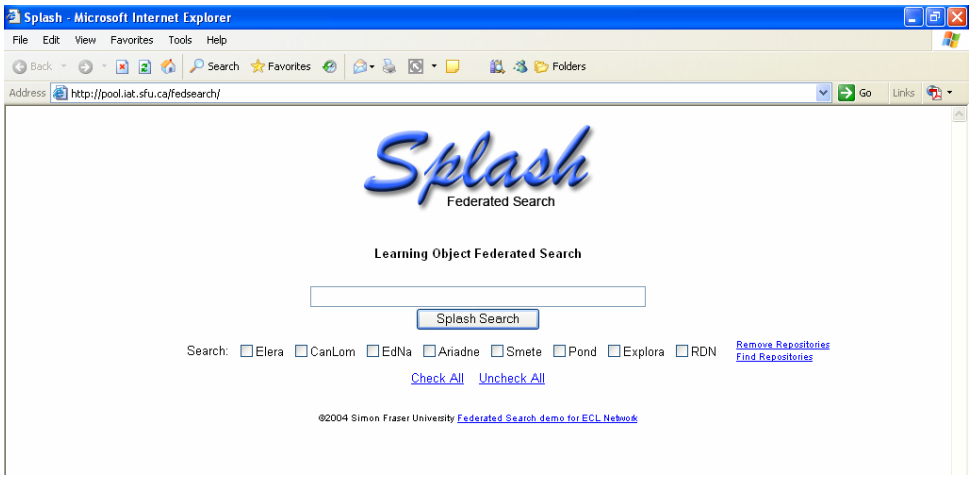
Table 1. POOL network nodes functionality
○ supported ● main functionality ▲specialized

The central website for the POOL project (www.edusplash.net) acts as a portal through which users can download the SPLASH application, seek support and most importantly hasten the search process between all SPLASH repositories and partner LOR projects. At the moment POOL has links with Elera, CanLom, EdNa, Ariadne, Smete, Explora and the RDN.

The POOL project uses the CanCore Protocol for all metadata stored within the repository(ies).

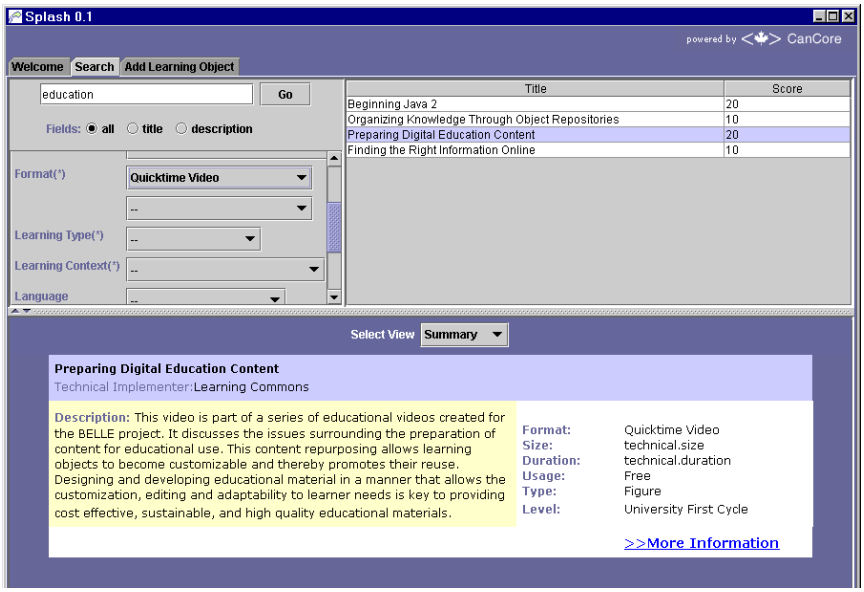
Screen Shots

Figure 7.4.2a – The SPLASH Federated Search Interface on the POOL website



[1] http://www.edusplash.net/upload/INET2-RichardsandHatalav2.doc

Figure 7.4.2b – The SPLASH Search Interface



7.4.3 Existing Implementations

At the time of writing this report the POOL website did not detail any existing implementations.

7.5 Eduplone

The Eduplone website is at www.eduplone.net

7.5.1 Background

Eduplone is a project, support by the European Consortium, based on Open Source Software.

7.5.2 Description

Eduplone is based on the popular open-source content management software, Plone (www.plone.org). Plone is a content management and publishing system, sharing the same qualities as Teamsite, Livelink and Documentum. The Plone software is built on top of the Zope architecture using the Python programming language.

Eduplone is a collection of OSS/FS products and customisations for the Plone software producing a LOR compliant with educational metadata and content packaging standards such as IMS.

The developers of Eduplone have produced a number of customisations of the Plone product built on the following four educational specifications:

- IMS Content Packaging
- IMS Learning Design
- IMS Question and Test Interoperability
- Learning Object Metadata

The Eduplone project promises to provide the following services to all instances of the repository:

1. Training Courses and Materials
2. Consultation
3. Study materials for guidelines on all aspects of learning objects including design, production and implementing standards
4. System development and maintenance

A demonstration installation of Eduplone can be found at the following URL:
http://demo.eduplone.net/

Screen Shots

Figure 7.5.2a – The Eduplone Interface

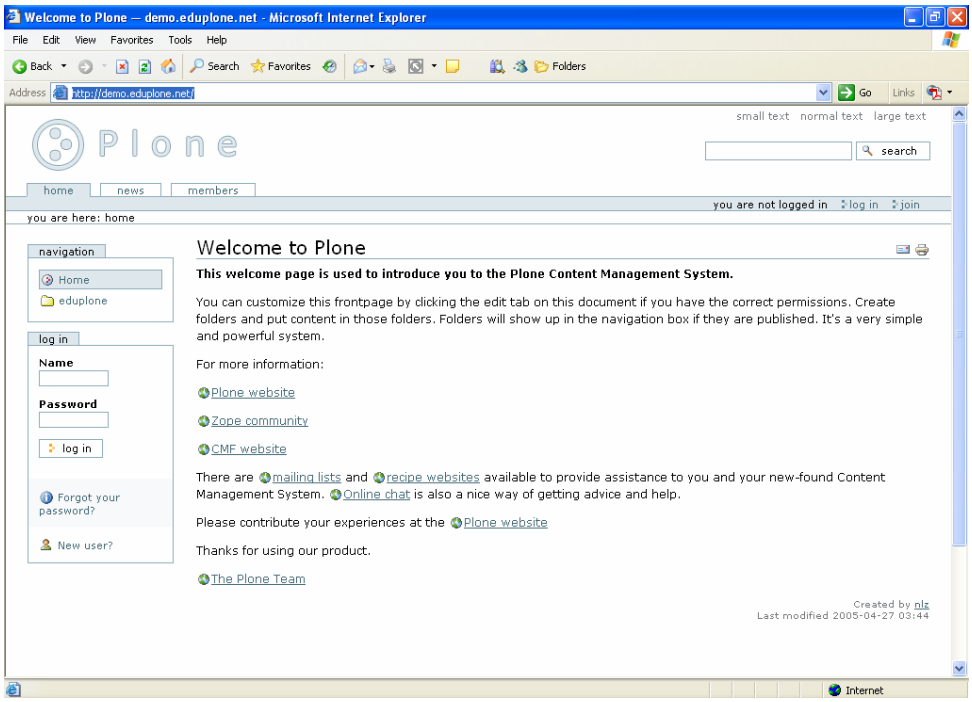


Figure 7.5.2b – The Eduplone User Area

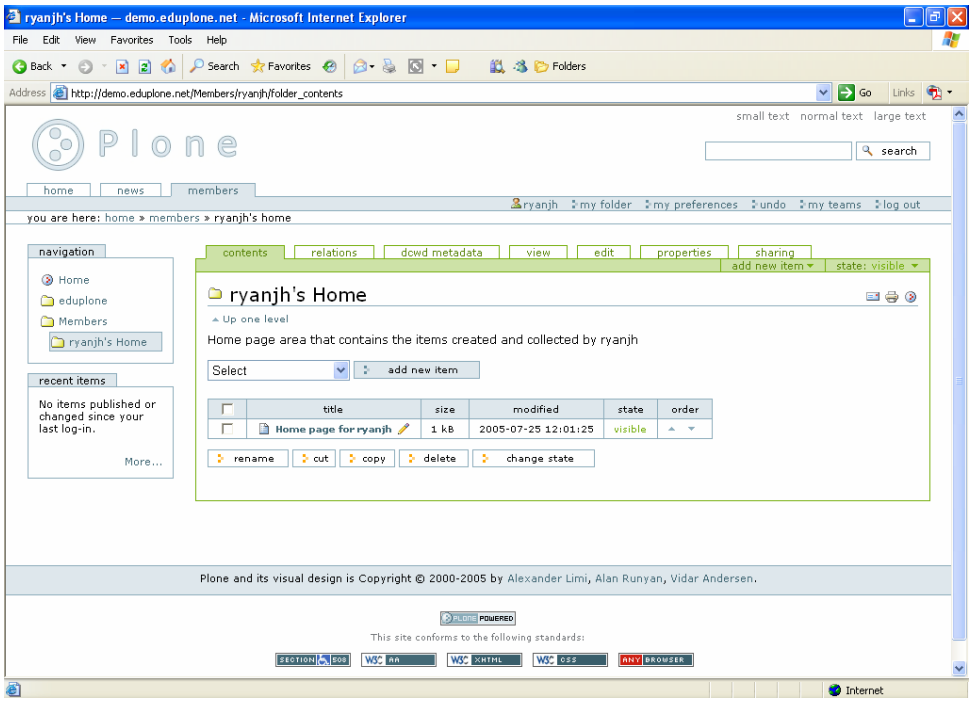
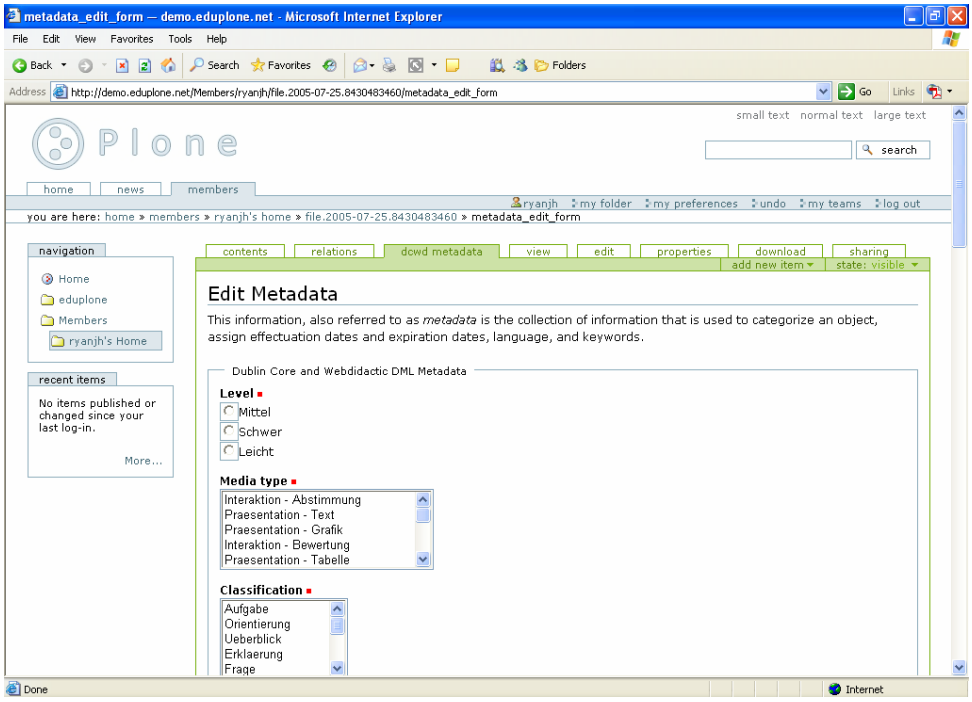


Figure 7.5.2c – The Eduplone Metadata (note, in German)



7.5.3 Existing Implementations

At the time of writing this report the Eduplone website did not detail any existing implementations.

7.6 eRIB

eRIB stands for eduSource in a Box (http://edusource.licef.teluq.uquebec.ca/ese/en/overview.htm)

7.6.1 Background

eRIB is supported by eduSource and is part of the CANARIE e-learning project in Canada.

7.6.2 Description

eRIB is a bilingual tool set that makes it easy for users to catalogue, upload and share learning objects, and to download, repurpose and reuse content in their own learning and teaching materials.

eRIB consists of an open-source database (eXist) with a built-in LOM (IEEE Learning Object Metadata) data structure and a set of tools to create, manage, upload, search and download resources.

The table below illustrates the various roles that a user can assume in the eRIB architecture.[1]

[1] http://edusource.licef.teluq.uquebec.ca/ese/en/roles_tools.htm

Table 2 Roels in eRIB

Actor	Role Tasks	Available Tools
Manager	A Manager installs, configures and maintains the repository for an organization.	• MANAGE eRIB • CONFIGURE Network
Builder	A Builder creates Learning Object Metadata records and adds them to the repository.	• CREATE Metadata • MANAGE eRIB
InfoSeeker	The InfoSeeker searches for LOMs and has several tools accessible to carry out simple keyword searches, narrow a search by putting constraints in the Advanced Search tool and to create and launch a Search Agent that will repeat a specific search at a certain frequency and then import resulting Metadata records directly into your eRIB.	• SEARCH • IMPORT (HARVEST LOMs) • ADVANCED SEARCH • SEARCH AGENT
Designer	A Designer creates, adapts and packages learning scenarios, called Learning Design (LD), according to international standards. UNDER CONSTRUCTION	• LD FIND AND SEARCH • LD Viewer and Editor • LD CONTENT PACKAGING
Publisher	A publisher adds digital rights metadata to a LOM by defining copyright and/or other restrictions. UNDER CONSTRUCTION	• Digital Rights Management Tool

Screen Shots

Figure 7.6.2a – The eRIB Interface Toolbar

Figure 7.6.2b – The eRIB Resource Creation Module

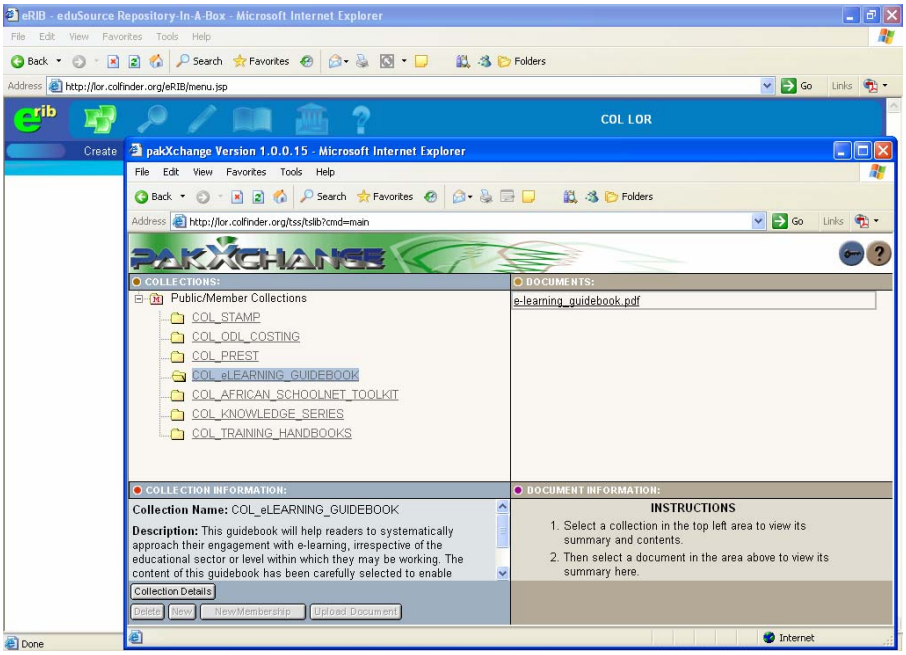
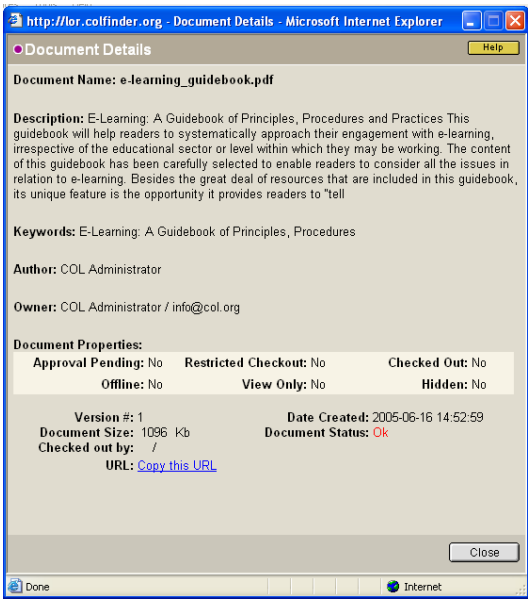


Figure 7.6.2c – The eRIB Metadata



7.6.3 Existing Implementations

The Commonwealth of Learning (COL) has created an OSS/FS LOR[1] using the eRIB software. The COLLOR demonstration system is configured to access the eduSource network of repositories. The system includes the ability to add information about the learning material ("learning objects") with "meta tags". These are IMS-compatible (IMS Schema 1.2.2) and the system is able to "federate-search" other repositories when the user is looking for learning materials.

7.7 The PLANET Digital Repository

http://ants.etse.urv.es/planetdr/

7.7.1 Background

The PLANET Digital Repository is a project supported by the University of Rovira I Virgili[2] in Spain, in association with the University of Pompeu Fabra, in Spain, and eduSource Canada.

7.7.2 Description

PLANET is built on top of the ECL (EduSource Communication Language)[3] which like JORUM is based on the IMS DRI (Digital Repository Interoperability) Specification[4].

> The ECL functions implemented by the URV's web services consist of search service, submit service and request service. There are different types of search, first, the quick search, that allows to ask for contents that verify in any field of the

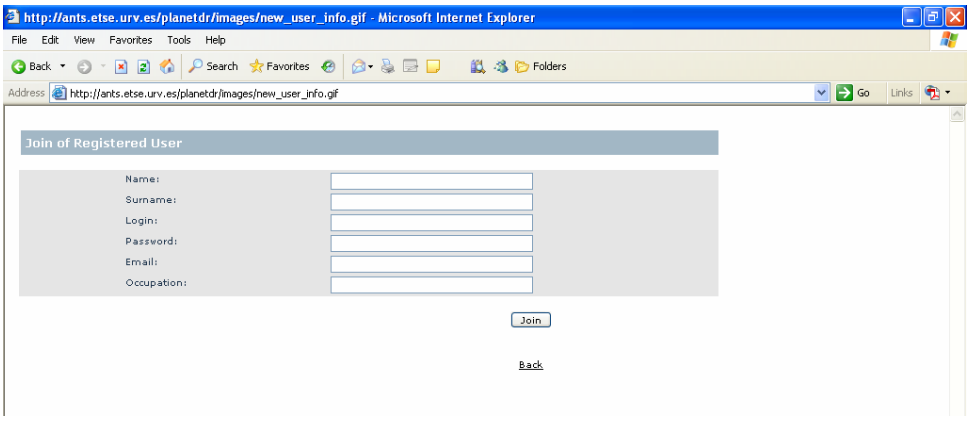
[1] Commonwealth of Learning http://www.col.org/lor/
[2] http://www.urv.net/home/home.htm
[3] http://ants.etse.urv.es/planetdr/info/info3.html
[4] http://ants.etse.urv.es/planetdr/info/info2.html

associated meta-data the wished word. The advanced search consists of two types: search by main meta-data category, where any field of a LOM meta-data category could be specified. The second type is accumulated search, that allows to make search on any field, linking together conditions of different LOM categories. It is possible to invoke the web services provided by the URV, or the web services implemented by any active content servers. It is also possible to invoke a federated search, (a simulated gather service) linking together the results of the requests of all the active content servers.[1]

The PLANET repository is java based and requires Apache Tomcat and MySQL.

Screen Shots

Figure 7.7.2a – Creating a New User



(Very few screen shots were available during the writing of this report)

7.7.3 Existing Implementations

At the time of writing this report the PLANET website did not detail any existing implementations.

[1] http://ants.etse.urv.es/planetdr/body.html

8. Selected Case Studies of Open Source Software Learning Object Repositories

8.1 Overview

The principal reason for selecting these repositories as detailed studies is for their perceived relevance to the JORUM. As stated earlier in the report, there are currently no known systems which replicate the JORUM repository's functionality, though some of these systems seem sufficiently similar to make a useful comparison.

In all the countries sampled here there are substantial investments in the e-Learning field as well as in the construction of the necessary technical and organisational infrastructure. This implies that similar questions are being asked and similar problems being solved in these countries. Although the actual usage of OSS/FS probably varies from country to country, all of them have substantial experience in the usage of OSS/FS.

8.2 City College Coventry Content Repository

We learned of this in-house developed system from a mail list trawl and it proved simple to obtain a demonstration version that could be installed on a local machine running Apache 2 with Perl installed. This is a small-scale development that acts as a searchable Web-based system to access learning materials produced for the National Learning Network (NLN)[1] to which Coventry College has access.

The specifications of the repository system are summarised here:

- The overall design and user interface of the repository emphasises simplicity and ease of use.
- It looks and feels like a search engine using a familiar web form, returning results to users in a relatively detailed manner compared to some systems. Full text searches are available, as well as phrase and Boolean searches.
- A Browse facility is available, with Subject categories down the left side of the page.
- Addition of content to the repository is as automatic as possible, using a simple web interface through which URLs, single files, or zipped collections of files can be uploaded to the repository as Content Packages. Content packages are generated by the Submitter themselves by using in-built features of the system.
- Tutors can integrate content with the institutional VLE (WebCT) through the use of hyperlinks, without having to separately download the content and then upload it into the VLE.

The content of the repository consists of the large amount of NLN learning materials (as Learning Objects) that the College subscribes to, which was previously added semi-automatically to a browsable web directory system, with different sections for each subject area.

The Coventry repository will be used in the JISC Distributed e-Learning programme, Regional e-Learning pilot projects[2] led by University College Worcester. The project is looking to produce a framework for promoting the use of shared digital resources in FE and HE institutions across the West Midlands, by making them available online using Digital Repository software.

[1] National Learning Network http://www.nln.ac.uk/
[2] "Promoting shared use of digital content across the region" http://www2.worc.ac.uk/wm-share/

8.2.1 Technical Aspects

The Coventry repository is built with OSS/FS components, using Apache2 server, Perl, and XML. The repository is written in Perl and uses the Gnome project LibXML2 and LibXSLT engines and works with these through CPAN modules. The repository is platform independent and is built around four basic components:

1. An indexer to harvest metadata and store it in a database
2. The database, which holds the keywords pointing to the content
3. A search screen to search the keywords in the database and return results
4. A content viewer to display content and menus, from which content could be used, or hyperlinks made

8.2.2 Metadata Harvesting

Only a subset of the LOM schema fields used by the NLN is used by this repository system; the indexer stores these in the database. The indexer currently takes around four minutes to index about 2000 Learning Objects.

The indexer is a command-line program that is run with arguments.

8.2.3 Database

The backend database is the native Perl DB_File system which interfaces to the BerkeleyDB. Rob Talbot (ILT Manager, City College, Coventry, UK), the repository designer, states that it would be simple to change to another database such as MySQL if there was an urgent need to do this.

8.2.4 User Interface

Users can either search the repository using a familiar web form or they can browse the resources, seeing a dynamically-generated subject listing. Results are returned as a link to the package along with a detailed textual description. On clicking the link, a Content Viewer opens. The navigation generated from the manifest is shown on the left and to view any page the user just follows the link from the navigation.
The whole package can be linked to from within a VLE, Word, PowerPoint or other document type by using the link provided which points to the database. This will open the Content Viewer. Tutors can also link to individual items on the navigation menu; each menu item acts as a URL to the resource.

8.2.5 Interoperability

The repository is known to work with content packages from the NLN and those produced with other packagers such as Microsoft Producer, CourseGenie and e-CAT. The application has been designed to work with **IMS Content Packages**, both reading from them and creating them. Metadata created conforms to a subset of UK LOM. Other metadata standards have not been addressed.

8.2.6 Workflow

Workflow is simple, but apparently adequate for the task. The following tasks are carried out during the workflow:

- Administrators drop existing content packages (e.g. NLN packages) directly onto the server.
- The indexer can then either be run manually to add them to the database, or be configured to run automatically on a nightly basis.
- The e-Learning Developers add metadata to the packages that do not have metadata already and, where possible, this is checked with Subject Matter Experts and amended if necessary.
- Content added by the Submitters are checked by Reviewers, who are members of the College Library staff.
- Submissions are flagged as pending until they have been reviewed.

8.2.7 API

There is a limited API. As a CGI application the search script, search.pl, can be called from any web form and the results returned. This makes it very easy to integrate with other web content. URL query strings can be hard coded to web pages to make persistent links, for example to a taxonomy category.

8.2.8 Future Developments

Any future developments depend on the continued funding and further uptake of this type of repository. The fact that the repository is expected to be used more widely may support further developments.

Examples of future developments include:

- Use of LDAP servers (based on CanCore principles)
- XQuery with native XML databases (e.g. the eXist database); the API in such a system seems extremely promising
- SOAP could abstract the interface from the information, so that a variety of interfaces to the same information could be developed

8.3 BCcampus – APOLLO/ CAREO

The experience reported to the JORUM team from BCcampus has been highly informative in our view of the issues raised concerning the use of OSS/FS for a learning object repository (LOR)[1]. The report here is granted by permission from the team at BC Campus.

To summarise, the use of OSS/FS needs careful planning and management, with a clear Business Case being identified. In this instance, the technical difficulties encountered were not trivial. These points apply equally to proprietary as well as OSS/FS.

This detailed study of the BCcampus experience is relevant to JORUM for the following reasons:

- The functional requirements for the LOR were similar to those currently adopted by JORUM (see Appendix for details)
- The practicalities of using, and managing, the OSS/FS development model for a rapidly developing application such as a LOR needs careful thought
- The effect of political commitment in preferring to adopt an OSS/FS solution, if one existed needs to be balanced

[1] Personal communication from Scot Leslie

8.3.1 Background

The project began in February 2004 to provide some form of LOR to house and share centrally-funded learning materials for BCcampus and OpenSchoolBC in British Columbia, Canada.

BCcampus was established in 2002, with a mandate to provide British Columbia learners with a web-based access point to online learning programmes and services. BCcampus is not itself another institution; rather it is a single, comprehensive point of access to online learning resources to cater for the large number of students involved in online learning in the province of British Columbia and outside. Beginning in 2004 it has brought online learners directly to the programmes and courses of British Columbia's higher education institutions.

BCcampus was able to build on the experiences of the CANARIE e-Learning programme with several OSS/FS repositories in service, especially the Campus Alberta Repository of Educational Objects (CAREO) repository. CAREO is an established OSS/FS repository system with several installs in the Canadian province of Alberta and elsewhere. Learning Commons Educational Object Repository[1] at the University of Calgary uses CAREO with currently over 4000 learning objects stored in the system.

8.3.2 APOLLO

APOLLO[2] (Academic Publishing for Online Learning with Learning Objects) was conceived as the University of Calgary's 2nd-generation LOR, based on the CAREO system. APOLLO was meant to be an improved, object-centred model and came from the Learning Commons group at the University of Calgary. This group were the creators of the CAREO repository software, already in use in British Columbia to support the revised 'Resource Pool,' and the new APOLLO software was already being used to support the SciQ[3] repository in Alberta. This is an early version of APOLLO in action. In addition, a Request For Proposal was issued in early July to engage a design firm to build the user interface to the repository software.

8.3.3 Technical Standards

Within the context of APOLLO and the requirements for their system, BCcampus were trying to implement importing and exporting of IMS Content Packaging, IEEE metadata, IMS DRI through the use of the Edusource Communications Layer, and workflow.

8.3.4 Technical Aspects

Actually finding the technical details for APOLLO has been problematic: an APOLLO weblog[4] is currently available and there are project Wikis[5] for APOLLO that point to a Macromedia Breeze presentation[6] that is quite informative.

APOLLO is based on WebObjects, an application server initially created by NeXT and now owned by Apple Computers. WebObjects are a widely-used, Java-based set of technologies, providing services for dealing with databases, generating interfaces, and managing low-level infrastructure. Licences cost $99 USD. Strictly speaking, APOLLO is not entirely OSS/FS in this regard.

---

[1] Learning Commons http://careo.ucalgary.ca/
[2] APOLLO http://apollo.ucalgary.ca/
[3] http://www.sciq.ca/
[4] APOLLO Weblog http://commons.ucalgary.ca/weblogs/apollo-dev/
[5] APOLLO Wiki http://commons.ucalgary.ca/cgi-bin/wiki.pl?APOLLO
[6] APOLLO presentation at: http://apollo.ucalgary.ca/project/events/NMCSpringConference04/presentation/

From a very high level view, APOLLO consists of two major WebObjects frameworks named CXFoundation and CXAppKit, which form a rich base of code that end-user applications draw upon.

CXFoundation contains the base functionality needed to run an APOLLO application. It handles access to repositories, digital objects (representations of resources, metadata, and assets), digital rights management, user services, plugin services, pasteboard services, directory services, virtual file systems, system communications, and workflows.

CXAppKit is an optional framework that takes care of "90%" of the tasks to build an application. It features an extension of WebObject's DirectToWeb technology for APOLLO, providing standard pages for searching and display search results, inspecting and editing objects, and many other tasks. CXAppKit accepts plugins that provide additional components and pages to be used with the DirectToWeb system.

The APOLLO designers placed particular emphasis on the move from a Metadata-centric to an Object-centric view of resources stored in the system. Another potentially useful design feature includes collections and nodes. Collections consist of a set of nodes that allow expression of relationships between groups of objects rather than having to manually create such collections: nodes can be represented in a variety of ways. This design feature is in the spirit of the Semantic Web and may have some resemblance to work done on Topic Maps. APOLLO was also designed to perform federated searches.

APOLLO does not refer to a single entity like CAREO; it is designed as a set of WebObjects Frameworks that any number of WebObjects applications can include and use. Each application can perform a different major function (e.g. generate presentations or Search and View), tied together into a “core” application whose job is to co-ordinate these applications.

8.3.5 Reasons for Choosing APOLLO

According to the BCcampus website, their recommendations for choosing APOLLO over a mix of 17 OSS/FS and proprietary systems were as follows:

1. As an open source option, APOLLO provides the flexibility, customizability and extensibility to deliver the solution as we define it. This is especially important in our multi-institutional settings where these needs are still emerging and will come out of continued consultation with system partners and end user

2. Adopting APOLLO offers the opportunity for inter-institutional partnerships … These partnerships provide yet another means for creating strong communities of shared interest

3. APOLLO offers the stakeholders the opportunity for a potentially more sustainable model of growth and deployment, with more control over the choices made in development and thus more control over costs. Enhancements can be paced over time to spread the costs and avoid the risks associated with one-time commercial licence fees and annual renewals. Implementing APOLLO will definitely have costs, but the stakeholders felt that maintaining some control of these costs was a key to developing sustainable business models

4. The implementation and development of APOLLO in BC offers the opportunity to foster local technical skills and expertise in learning object technology, and further enhance BC's and Canada's growing reputation and skill base in this burgeoning area[1]

[1] Final Product Evaluation available from: http://www.bccampus.ca/Page227.aspx

8.3.6 OSS/FS Problematic Aspects

Some general reflections are offered here for the benefit of the wider e-Learning community. The reflections are based on Leslie (op cit)[1], concerning using the OSS/FS development model for a LOR.

Software development work

It is easy to underestimate the amount of coding required to adapt existing applications. Although a great deal of experience from the University of Calgary was carried over into the APOLLO project there was substantial new code to be written. Time scales are usually limited and difficulties will inevitably arise, especially where new functionalities (e.g. Workflows) are being devised.

Release of code

If a project is to use the OSS/FS model then it needs to fully commit to that model. Code needs to be released to the wider community as soon as possible so that as many eyeballs can examine the code for bugs etc. Code needs to be downloadable even when not complete.

Project management is important

When problems arise (e.g. code not being ready on time, lines of responsibility unclear and so on) then standard project management techniques and organisational structures are useful.

'Political' preferences

Although the will to consider and use the OSS/FS model is very important in order to overcome the sheer inertia of the more familiar proprietary model, a full evaluation of all the risks involved is necessary. A cost-benefit analysis of opting for an OSS/FS solution, compared to a proprietary one, in any particular circumstances, is useful.

Experience shows that sufficient technical expertise needs to be available within any project, especially if there is the need to assess the viability and functional capabilities of externally supplied software. This was the case with APOLLO.

Funds were limited, and the development time-scale very short; two important factors that seemed to have led to the problems met later on in the project. Another problem, at least as significant, was the belief that there already existed an OSS/FS solution that could simply be adopted and implemented. Numerous problems were encountered in the attempt to use APOLLO; the principal technical ones being to underestimate the amount of development work needed. Also, insufficient technical expertise was available within the project, which made it difficult to assess the viability of the presented code described as "APOLLO". By January 15 2005 BCcampus cancelled its involvement with the University of Calgary and the APOLLO project. The code was not released into the open as OSS/FS at the start of the project by the University of Calgary, and instead the project team planned to release the code as OSS/FS once it was at a suitable stage. In these terms, the APOLLO product was not truly OSS/FS when BCcampus started to work with it; this meant that potential help from the OSS/FS community was unavailable.

In short, too many risks were carried by BCcampus which led to them eventually abandoning their plan to use APOLLO and to seek another supplier.

[1] Personal communication from Scott Leslie

8.4 Connexions

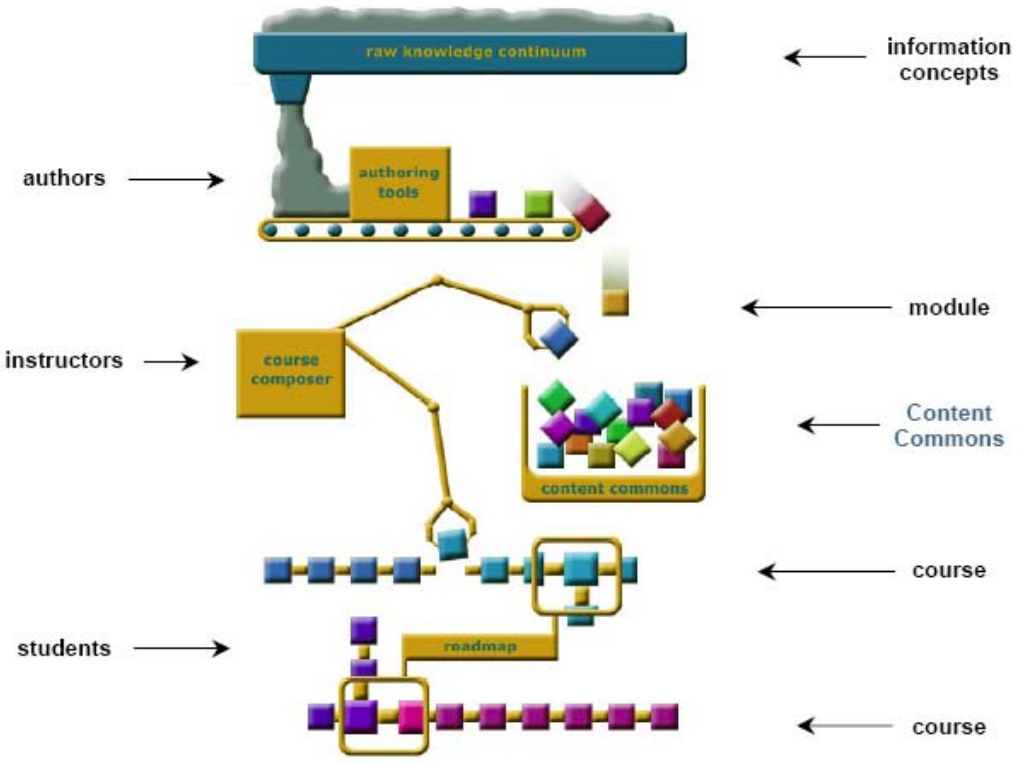
8.4.1 Introduction

The main aim of the Connexions project is to develop an environment for collaboratively developing, freely sharing, and rapidly publishing scholarly content on the Web. The LOR was built on top of the OSS/FS Plone[1] software discussed in previous sections of this report. The Connexions project then developed additional tools for the repository to allow users to author learning objects through the software its self. All content stored within the Connexions repository is free to download and reuse under the Creative Commons "Attribution" Licence.

> Connexions is a web-based document creation and management system for education and research materials. Connexions promotes collaboration among authors, instructors, and students – across disciplines and between institutions. And it also enables an individual to self publish documents. All of this is accomplished using open-access software tools and free-use material through the Creative Commons Attribution license.[2]

The diagram and text below help to illustrate the architecture of the Connexions project:

Figure 8.4.1a – Connexions Architecture[3]



The Connexions strategy relies on a global community of authors producing reusable modules of content. These are created using the connexions authoring tool or a suitable alterative.

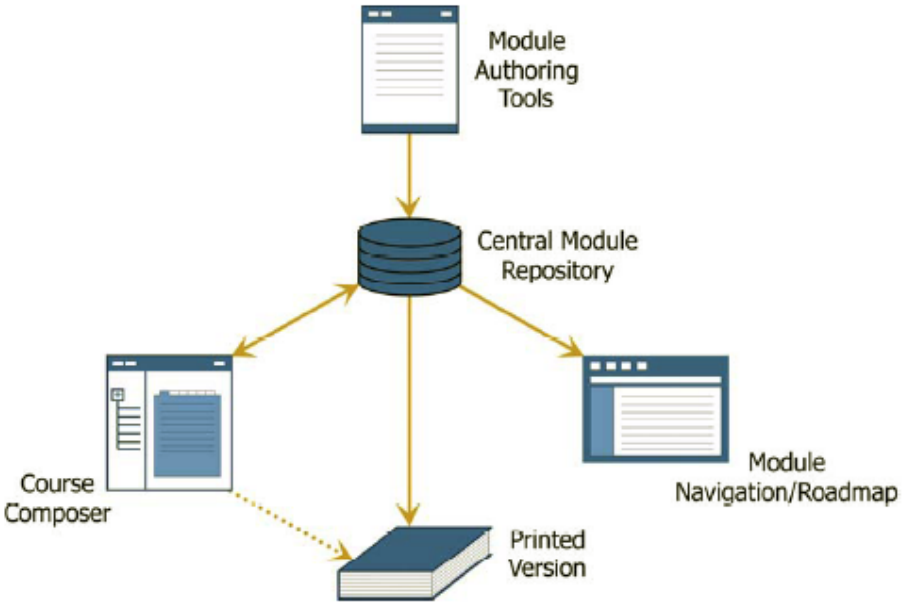
[1] www.plone.org
[2] http://cnx.rice.edu/help/faq/view
[3] http://cnx.rice.edu/aboutus/publications/ConnexionsWhitePaper.pdf

> Modules can be imagined as special Web pages with hyperlinks pointing to prerequisites, applications, and supplementary material.[1]

Modules are then stored in the Connexions repository known as the 'Content Commons'. Modules are uploaded to the Connexions repository through the website (http://cnx.rice.edu/). Instructors use the 'Course Composer' to create customised courses based on modules stored within the Content Commons repository. These courses can then be delivered through the web, presented in the classroom or simply printed off for students. Learners can also use software developed by the Connexions project called 'roadmap' to access resources online and track their progress.

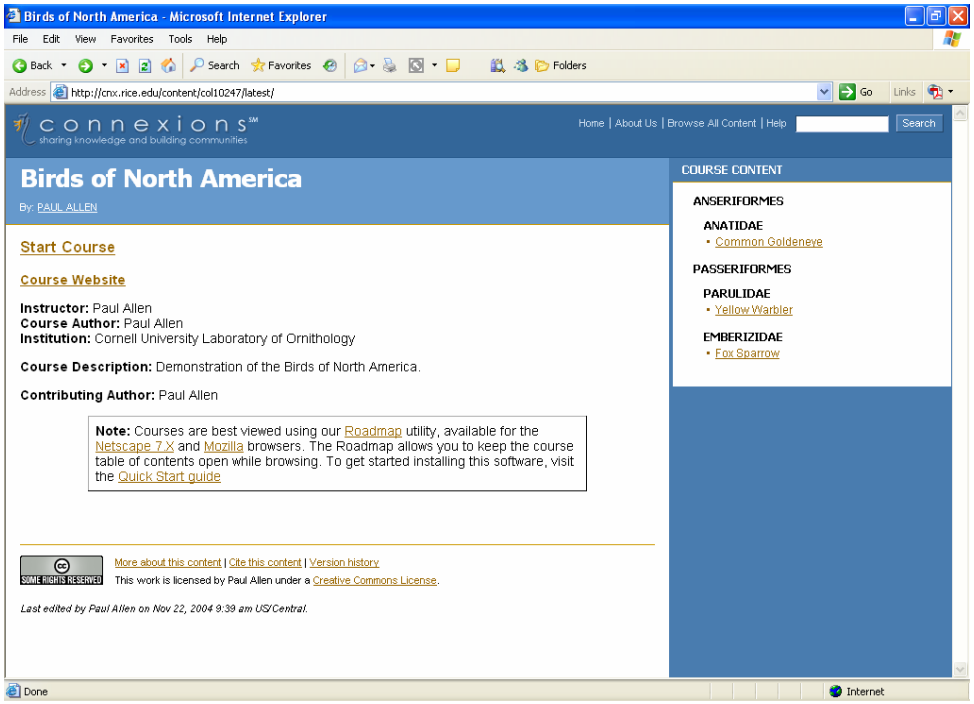
> Students and other learners access Web courses or the Content Commons directly, using special visualization and navigational tools designed to highlight the non-linear "Connexions" among concepts both within the same course and across courses and disciplines.[2]

Figure 8.4.1b – Connexions Software Tools



[1] http://cnx.rice.edu/aboutus/publications/ConnexionsWhitePaper.pdf
[2] http://cnx.rice.edu/aboutus/publications/ConnexionsWhitePaper.pdf

Figure 8.4.1c – Example of a Connexions Course



8.4.2 Background

The Connexions project was born in 1999 and is supported by Rice University in Houston, Texas, USA.

8.4.3 Current Developments

At present there are 2300 modules and 35 courses stored within the Connexions repository and in current use.

At the moment their content collection is focused on two areas: Music and Digital Signal Processing. This strategy is mainly to promote community building within the Connexions user group. As the project expands content collection will expand into new fields.

Existing contributors include:

- Rice, Illinois, Ohio State, Georgia Tech, Polytechnic, Michigan, Cambridge (UK), NTNU (Norway)
- National Instruments & Texas Instruments
- Botanical Research Institute of Texas
- American Museum of Natural History
- Creative Commons
- UNESCO (with MIT OCW and CMU)
- American Society for Engineering Education

Currently when modules are imported into the Connexions repository they must be in CNXML which is the format chosen by the project. To reduce the technical knowledge required by authors, the project is developing importers for commonly used formats like Microsoft Word, and XHTML that will convert documents into CNXML. For the more technical author, the

project is also working with Disruptive Innovations[1] to build a fully functional standalone XML editor that will be cross platform and open source.

To promote communication and support within communities of users Connexions is developing:

- Discussion forums for the site and in workgroups so that authors, instructors and users can better collaborate
- Author profile pages
- Community pages

8.4.4 Future Plans

The connexions team are planning to move the project towards the following directions/goals:

- Full open-source software release to allow anyone to set up a collaborative open-content site
- Microsoft Word importer and Enhanced Edit-In-Place functionality
- Author profile pages
- Improved translation support to better enable multilingual content
- Site forums
- Workgroup discussion forums
- Demonstration of courses with modules from multiple repositories
- XHTML/MathML importer
- Content rating system
- Content exporters
- Open-Source XML editor
- Open community page feature for beta testing
- Spell checker
- Federated repository integration[2]

8.5 Conclusions

Although there are no off-the-shelf OSS/FS LOR currently available for the JORUM project to use at present there are some promising developments occurring in the repositories reviewed in this section. The following comments summarise conclusions about these repositories.

There are some important lessons for the JORUM service from the BCcampus experience with APOLLO. BCcampus actively considered a range of possible repository systems and had a very similar set of technical requirements to JORUM. These included content storage features such as IMS Content Packages to store multiple files, metadata repository as well as the learning objects themselves. Similar metadata schemas, with the ability to support roles, permissions and the various workflows were needed to process the objects. Interoperability with other learning object repositories, archives and library systems was required, including OAI-PMH compatibility. The fact that they chose to follow the OSS/FS path and were disappointed is clearly relevant to the JORUM service, now or in the future. The problematic issues they faced when trying to adopt the OSS/FS model are outlined in Section 6.2.2 and JORUM would do well to take note of the lessons learned.

Are there any clear lessons for the JORUM project from the experiences of BCcampus? It would be necessary for JORUM to follow standard procedures for procuring new software, including a thorough risk analysis, as already carried out for the current system procured under EU regulations. As regards using off-the-shelf OSS/FS for the JORUM service, this is not currently relevant as a proprietary, standards-based system has already been procured for the start of the service. However, development of an OSS/FS solution is a possibility for

[1] http://disruptive-innovations.com/
[2] http://cnx.rice.edu/aboutus/future

the future, given the wealth of expertise worldwide and domestically. The relevant resources would need to be made available, both to develop or modify the code and, equally important, maintain and support this software over the longer term.

City College Coventry seem to be demonstrating how a local, small-scale system can be applied in the UK e-Learning context to effectively support academics using quality-assured learning objects from the NLN. This system has some of the basic features that a larger-scale solution such as JORUM would require (e.g. the use of IMS content Packaging for their objects, indexing of objects using metadata and a simple, but effective workflow).

The Connexions project was included as a case study because it is an excellent example of a learning object repository built from a customised generic digital repository as defined in section 4.1.

By incorporating a proven open source content management system like Plone, the Connexions project inherited a number of pre-existing advantages.[1]

- Built-on functionality generic to all repository applications Plone is suitable as a document publishing system and a groupware tool for collaboration between separately located entities, which is essentially what the Connexions project was trying to achieve.
- The Plone Team includes usability experts who have made Plone easy and attractive for content managers to add, update, and maintain content.
- The Plone interface has more than 50 language translations, and tools for managing multilingual content.
- Plone carefully follows standards for usability and accessibility. Plone pages are compliant with W3C's AA rating for accessibility, in addition to using best-practice web standards like XHTML and CSS. This is essential by law for any project offering an educational service.
- Plone's system is built around a user / role / permissions workflow model which is essential for a learning object repository
- There are close to a hundred developers in the Plone Development Team around the world, and a multitude of companies specializing in Plone development and support.
- There are many add-on products for Plone that add new features and content types. In addition, Plone can be scripted using web standard solutions and Open Source languages
- Plone can interoperate with most relational database systems, open source and commercial, and runs on a vast array of platforms, including Linux, Windows, Mac OS X, Solaris and BSD
- Plone has an integrated and powerful search engine

By examining the Connexions case study in detail there are valid lessons that can be learnt:

- Interoperability is at the core of both the Connexions and JORUM strategy. However, Connexions adopt more generic interoperable standards such as XML and focus more on building a suite of open source tools to complement each stage of the learning object life cycle.
- There are a number of open source solutions that could be classed as repository applications e.g.: Document Management Systems, Content Management Systems, and Knowledge Management Systems. The Connexions project was able to ignore the labelled terminology of their chosen OSS/FS system and take advantage of the pre-existing functionality that has helped to make the project so successful.
- The Connexions project is open to the entire US academia; therefore the target audience is extremely vast. The strategic move to focus on a very limited segment to increase content collection, community building and communication is one that will surely be adopted by many existing LOR projects.

[1]Based on information extracted from the Plone website www.plone.org

- By building open source tools to assist with the whole learning object lifecyle, from creation, through to repurposing, the Connexions project offers a single gateway for their myriad users.
- The production of small, reusable modules removes the requirement for complex disaggregation functionality in the system, and promotes course building of infinite combinations.

9. Web References

9.1 General

JORUM Scoping and Technical Appraisal Study Vol 1 Overview and Recommendations http://www.jorum.ac.uk/research/publications.html#scope

JISC ELF http://www.elframework.org/

Open Source: national frameworks conference presentations available at: http://www.oss-watch.ac.uk/events/2005-01-20/

David Wheeler article 'Why Open Source Software …' available at http://www.dwheeler.com/oss_fs_why.html

The Open Source Initiative - A non-profit corporation founded by Eric Raymond to promote Open Source through OSI certification http://www.opensource.org/

Listing of OSS/FS licences http://www.opensource.org/licences/

Writings of OSS/FS 'names', like Eric Raymond available at: http://www.catb.org/~esr/writings/cathedral-bazaar/

9.2 Guides

OSI Guide to Institutional Repository Software 3rd August 2004 Edn. available at: http://www.soros.org/openaccess/pdf/OSI_Guide_to_IR_Software_v3.pdf

9.3 Recent Reports on OSS/FS

BECTA report available at: http://www.becta.org.uk/corporate/publications/documents/BEC5606_Full_report18.pdf

FLOSS survey http://flosspols.org

OPEN SOURCE SOFTWARE Use within UK Government Oct 2004 http://www.govtalk.gov.uk/documents/oss_policy_version2.pdf

OPEN SOURCE SOFTWARE http://www.dti.gov.uk/bestpractice/assets/oss.pdf

OGC "Proof of concept" final report Oct 2004 http://www.ogc.gov.uk/embedded_object.asp?docid=1002367

CANARIE report 'Learning Object Repositories: Deployment and Diffusion' D. MacLeod, February 2005 http://www.canarie.ca/funding/elearning/2005_LOR_report.pdf

The value of Open Standards presentation by Brian Kelly (UKOLN) available at: http://www.oss-watch.ac.uk/events/2003-12-11/open-standards.pdf

9.4 OSS/FS Organisations

OSS Watch http://www.oss-watch.ac.uk/

The Apache Software Foundation (ASF) http://www.apache.org/foundation/

Open Access Initiative http://www.soros.org/openaccess/read.shtml

The ethics of Free Software (FS) http://www.gnu.org/philosophy/philosophy.html

9.5 Example Systems Using OSS/FS

Bodington http://bodington.org/index.jsp

Moodle http://moodle.org/

Sakai http://www.sakaiproject.org/cms/

uPortal at http://www.uportal.org/

Plone http://plone.org/

Fedora http://www.fedora.info/

DSpace DSpace http://dspace.org/index.html

Wikipedia at: http://wikipedia.org/

EPrints http://software.eprints.org/

9.6 OSS/FS Learning Object Repositories

DLearn at http://www.dlearn.arizona.edu/index.jsp

Connexions http://cnx.rice.edu/

CAREO http://www.careo.org/

Commonwealth of Learning http://www.col.org/lor/

TILE http://www.inclusivelearning.ca/tile/index.html

9.7 OSS e-Learning 'Tools'

RELOAD website http://www.reload.ac.uk/

TOIA website http://www.toia.ac.uk/

An eduSource Communications Layer(ECL) overview available at:
http://lore.iat.sfu.ca/ecl/#LORNET

Open Source Initiative (http://www.opensource.org/) - A non-profit corporation founded by Eric Raymond to promote Open Source through OSI certification.

http://www.opensource.org/licences

SourceForge (http://sourceforge.net/) - The self-proclaimed world's largest repository of Open Source code and applications.

Fedora Version 2.0 Open-Source Repository http://xml.coverpages.org/ni2005-03-18-a.html#deployment

http://www.egovos.org/ US Center of Open Source & Government

Appendix: BCcampus

For further information see the BCcampus FAQ website at
http://www.bccampus.ca/Page410.aspx

BCcampus was established in 2002, with a mandate to provide British Columbia learners with a web-based access point to online learning programs and services. The BCcampus initiative brings together existing online resources to ensure maximum convenience and value for publicly-funded online education services. It also fosters collaborative design and engineering of new courses, services and tools to reduce duplication and incorporate best practices into online instruction. In addition, it ensures that educators have the technical support and resources they need to provide quality online instruction.

BCcampus is not itself another institution, rather it is an online service that connects to institutions that will provide many of the services previously available through the (Open Learning Agency) OLA. It is an attempt to provide a single, comprehensive point of access to online learning resources to cater for the large number of students involved in online learning in the province of British Columbia and outside. Beginning in 2004 it has brought online learners directly to the programmes and courses of British Columbia's higher education institutions.

A List of Core Requirements for a BCcampus/OSBC Repository

Whatever system/method we choose must implement the following capabilities:

- Browsing and searching
- Browsing for objects through a variety of taxonomies and vocabularies (e.g., not only "by subject" but also "by grade")
- Simple and advanced searching capabilities, ideally including the ability to further refine searches, save searches, and share the results of searches outside the confines of the repository

Metadata Schemas and Cataloguing

- Potentially support at least two different metadata application profiles (post-sec and K-12)
- Batch uploading of both metadata and objects, along with automated schema validation for the metadata and link checking for external objects
- Metadata authoring tool provides some degree of automated metadata creation
- Support roles, permissions, and workflow for object submission and metadata cataloguing

Content Storage

- Provide both a metadata repository and a content store for the objects themselves to be uploaded to
- Handle everything from single media assets in multiple media formats, assets, or objects comprised of multiple files (e.g. PowerPoint Web exports—"Compound Documents") to IMS Content Packages

Aggregate Objects and Aggregation

- Ideally support aggregate/traversable objects
- Ideally support some end—user tools to aggregate new objects
- At base provide for personal collections (bookmarks) of objects

Community Features

- Ideally support new object "wish lists" and other discussion capabilities
- Provide means for displaying actual object downloads, and point to instances of objects being reused
- Possibly support other community—building features and evaluative metadata

Digital Rights Management

- Support basic DRM capabilities, including the ability to identify resources by publisher, identify user by organization, and restrict access to objects based on business agreements between publishing and consuming organizations
- Ability to assign an individual account to one or more roles
- Ability to assign an individual object to one or more collections (system collection, not personal collection)
- ability to assign rights and privileges to roles
- ability to manage DRM based on roles and collections (i.e., not by account and object)
- ability to track and account for "downloads" or "uses" of objects and report back to object publishers

Integration and Interoperability

- Interoperability with other learning object repositories, archives and library systems, including:
    - support harvesting via the OAI-PMH
    - support for Digital Repository Interoperability
- Provide a public API, Web services, or other means of extending functionality or integrating with other applications

Administrative and Other Functional and Non-functional Requirements

- Administrative functions through a Web client
- Usage information on the repository as a whole, and specific usage details on objects to end users
- Ability to work with an external LDAP directory for both authentication and authorization information
- Scale to handle tens of thousands of objects and multiple thousands of users (not including students accessing the content store)
- Must be clear that the data in the system can be migrated through a standards-based process out of this particular software into another system at a later date, if need be

Look and Feel

- Provide the ability to at least brand the system, and ideally generally customize the look and feel of the repository
- Ideally, based system can provide some personalization (e.g., different default points on taxonomies for browsing, portal-like features)
- Conform to basic accessibility standards (W3C WCAG Priority Level 1)
- Support the Unicode character set[1]

[1] BCcampus Core Repository Requirements available from: http://www.bccampus.ca/Page155.aspx