

Open Source systems in the library and on the Net

By

Dr. E. Rama Reddy,
Librarian, University of Hyderabad
Hyderabad – 500 046
errlib@uohyd.ernet.in

1. Open Source Software:

There are several competing interpretations of open source software. The way to describe is: It is software created by programmers who want to share their source code, which is part of a program that is readable by humans with anyone who might find the program, or a variation of the program, useful. In words of Richard Poynder, the term open source refers to software in which the source code is freely available for others to view, amend and adapt. It is maintained by a team of developers cutting across the institutional and national boundaries.

According to the www.oss4lib.org, the clearinghouse for information on open source development within the library community, open source software means many things.

- It is typically created and maintained by developers crossing institutional and national boundaries, collaborating by using internet based communications and development tools;
- Products are typically a certain kind of “free”, often through a license that specifies that applications and source code are free to use, modify and redistribute as long as all uses, modifications and redistributions are similarly licensed;
- Successful applications tend to be developed more quickly and with better responsiveness to the needs of users who can readily use and evaluate open source applications because they are free;
- Quality, not profit, drives open source developers who take personal pride in

seeing their working solutions adopted;

- Intellectual property rights to open source software belong to everyone who help to build it, simply uses it, not just the vendor or institution that created or sold the software.

A more succinct definition from www.opensource.org claims that “Open source promotes software reliability and quality by supporting independent peer review and rapid evolution of source code. To be certified as open source, the license of a program must guarantee the right to read, redistribute, modify, and use it freely”. Software that is operated under the open source label has to fulfill several criteria. The most important of these are:

- Can be circulated freely. Anyone may use open source software and pass it on as often as he or she wishes
- Availability of the source code. The software suite must contain the source code or this must be available to a freely accessible location.
- Modifications to the source code. The source code may be adapted to individual requirements and be passed in this modified format.

There are many open software projects that libraries use, often without even realizing they are using, open source software. These programs are often Internet based and have a long history of successful use. Examples of such projects are BIND, Perl, Apache and Linux. Bind, the Berkeley Internet main domain package, is software that allows one computer to find another without having to know its unique Internet protocol numeric address, In other words it translates 10.0.1.100. in www.mydomain.com and vice versa. It has been in use since 1984. Perl is a widely used scripting language and responsible for much of the active content of the WWW. It is commonly used for CGI scripting, accessing databases, text processing, XML processing, system administration, web transactions and many other activities. It has been in use since 1987. Lastly apache, an open source web server, the number one web server used on World Wide Web. Today, more than fifty percent of www web

sites are on apache server.

2. Advantages of open source software:

Motivations for using and developing open source software range from philosophical and ethical reasons to pure practical issues. The first perceived advantage of open source models is the fact that open source software is made available gratis or at a low cost. But this characteristic is not exclusive to open source model as several proprietary software products are made available in similar ways (ex. Microsoft Explorer).

1. The availability of the source code and the right to modify. It enables the unlimited tuning and improvement of a software product. It also makes it possible to port the code to new hardware, to adapt it to changing conditions, and to reach a detailed understanding of how the system works. Source code availability also makes it much easier to isolate bugs, and to fix them.
2. The right to redistribute modifications and improvements to the code and to reuse other open source code permits all the advantages due to the modifiability of the software to be shared by large communities. The right to redistribute which is universal and cannot be revoked is what attracts a substantial crowd of developers to work around the open source software projects.
3. The right to use the software in any way. This ensures a large population of users, which helps in turn to build up a market for support and customization of software, which can only attract more and more developers to work in the project. This in turn helps to improve the quality of the product and improve its functionality.
4. There is no one with the power to restrict in a unilateral way how the software is used, even in a retroactive way. For instance, a proprietary software vendor may decide not to upgrade some software product for some old platform and the customers can only stick to the old version of the software or switch to another product. But with OSS the customer can find some vendors to provide the upgrades.

5. There is no single entity on which the future of the software depends. If a proprietary software vendor closed doors or decides to discontinue development of the product, no one has the right to take the program and continue development on it, thus effectively killing its usability in the market. But OSS effectively protects against this, because if the group or company that originated the code decides to stop development, it is always possible to fund another software group to continue the maintenance and improvement with legal or practical limitations.

3. Disadvantages of the software:

There are some potential problems that one should be aware of when evaluating the feasibility of using open source software.

3.1. Compatibility.

It's not always easy to get open source software to work with other applications. More and more institutions are requiring nonprofits to use databases to track information, and those databases must operate with the standard set by the institution. Often, that standard is the office suite most often used, Microsoft Office, which isn't compatible with most open source programs.

3.2. Lack of control and Responsibility

The decentralized distribution of OSS process is a disadvantage in some contexts, particularly for fixing bugs and problems, which become the responsibility of user if the OSS community doesn't mobilize to address the problems. The lack of central controlling hand in OSS development also means that progress can be chaotic and undirected. Organizations should have a very conservative approach in selecting the code and implement that which is used by large numbers of people.

0.1. Technical support

Support may be sometimes difficult to understand because it is frequently aimed

at developers and not end users. Lack of proper guidance on many open source software, there is lack of user documentation because development of the code enthuse many as compared to documentation.

Libraries and librarians have for many years been using, creating and sharing software, scripts and batch files. Recently, with the growing popularity of the open source movement, some librarians have started extending patronage to open source software.

The first well known advocate for open source software in the library community is Daniel Chudnov who published an article entitled “open source software: the future of library systems” in library journal. This article listed many library related open source projects and gave a list of URLs so that librarians interested in learning more could research the open source phenomenon for themselves. The website www.oss4lib.org email discussion list is maintained by Daniel Chudnov.

4. Overview of open source software in Libraries:

4.1. Koha

Koha (www.koha.org) is considered as the first open source library automation system started by the Horowhenua Library Trust, a New Zealand consortium and Katipo communications. It runs on the Linux operating system in conjunction with the Apache web server, uses the popular MySQL open source database management system and is written in Perl. It includes all the features of any proprietary software. However, efforts are on for support for import and export of MARC record, Z39.50 client and server modules, CIP or NCIP support, and authority control. A group of volunteer programmers has been working on extending Koha's capabilities to include these essential features and others. Efforts are also underway to translate the system into several languages. It is fully Internet based and customizable. It can be downloaded from the site www.koha.org. Koha is already implemented in the Jorowhenua District Library and a number of school libraries in Canada.

4.2. Learning Access ILS

Learning Access Institute, a non profit organization dedicate to developing low cost strategies to increase digital access offers an open source automation system called the Leaning Access ILS (previously known as OpenBook). The development of Learning Access ILS was sponsored by the now defunct Technology Resource Foundation (TRF). Willem Scholten, the executive director of Learning Access Institute, was also associated with TRF. The Learning Access ILS is a fully Open Source and GPL licensed software application aimed at automating small to medium public and school libraries. It integrates the latest of Internet and Web technology. One of its main objectives is to provide a low-cost and flexible system that adheres to most of the common library standards for organizing bibliographic data. The Learning Access ILS consists of three modules: the patron or user module (OPAC), the cataloging module and the circulation module. In future releases it may also include an acquisition module. It runs on Linux, but can easily be ported to Windows NT/2000. It uses the Apache Web server and relies on MySQL as its database engine. The user interface modules are written in PHP, a Web-oriented programming language. The majority of the remaining system programming is done in Perl. Its usage is currently limited to initial set of libraries that work directly with the Learning Access Institute. The organization will release its system for unrestricted download once it's reached a satisfactory level of stability and maturity. To keep abreast of its developments, www.learningaccess.org/website/techdev/ils.php can be explored.

4.3. Avanti

Avanti project is started by Peter Schlumpf in 1998. It is written in Java and is platform ndependent. It includes its own database management. It was released on 14th January 2003, Avanti MicroLCS version 1.0 is a next-generation library system for the Java platform. Development of Avanti project can be monitored at <http://www.avantibrarysystems.com>.

5. Conclusions:

At this stage, there is a great deal of possibility for library specific, open

source software. There are several individual projects that are attracting the attention of many in the library community. Given the sharp rise of open source software, it seems just a matter of time before library specific open source projects overwhelm the community.

But it also raises a number of challenges. The apprehensions of continued technical support, learning, training and skills used to implement automated system are yet to be addressed. Though the cost of implementation is substantially lower than proprietary software, they have to carry the burden of development themselves or turn to a commercial vendor to mould the product to their needs.

While OSS struggle to offer only core modules, the overall scenario is changing beyond traditional functionalities. The Libraries demand services integrating Internet and Web technologies. Most of all, library systems must deliver full-fledged content, not just be finding-aids for physical collection. But in spite of the challenges, libraries should consider the capabilities of open source systems and evaluate the merits of the features, reliability and support.