

“Open Source’ Integrated Library System Software

By Richard W. Boss

A number of public libraries have been investigating “open source” integrated library system software. Their motivations appear to be both financial and a desire to tailor a system to more closely meet their requirements than the proprietary products allow.

The term “open source” refers to software that is free and that includes the original source code used to create it so that users can modify it to make it work better for them. It also includes the right of redistribution; therefore, there may be open source and proprietary products that are based on open source products. While the software may be free, a developer or distributor may charge for services, including special programming, installation, training, technical support, and hosting services for libraries that do not want to implement and maintain an in-house system. Some distributors of open source software are for-profit organizations and may charge prices for their services comparable to those charged by proprietary software vendors.

It is difficult to determine how many libraries are using open source integrated library systems because many of the libraries that have downloaded the software decided not to use it. The number of libraries and consortia that have support contracts for open source software is estimated to be 300 worldwide.

The perceived advantages of open source software are:

- *Ability to tailor to fit local needs*

The availability of the source code means that a user can modify and enhance the software to more closely fit its own needs. Unlike with proprietary products, the development priorities are set by the user, not a vendor. The user is also able to set its own priorities for bug fixes.

- *No restriction on use*

Unlike proprietary software, there are no contractual restrictions on how the software is used. While some developers use the GNU General Public License that assures users that they have the right to distribution and those to whom they distribute also have the right to modify and distribute, other developers merely declare that their software is in the public domain. A subsequent user may, therefore, decide to protect the enhancements that it makes by copyrighting them. The GNU General Public License is, therefore, preferable.

- *Low cost*

There is no charge for the software itself, therefore, the capital outlay required by proprietary software is avoided. The major costs are ongoing development and maintenance. If the number of users is large, and they share their efforts, each user's cost is reduced. However, if the number is small or a user does a lot of tailoring to fit unique local needs that are not shared by other users, the cost can escalate.

The potential disadvantages of open source software are:

- *Unanticipated cost*

Open source software tends to be less complete than proprietary software. Acquisitions, serials control, and interlibrary loan modules are often missing or incomplete. A library may find that it needs to do a great deal more work than anticipated to adapt the software to local needs. For example, one mid-size public library had to budget \$100,000 to modify the acquisitions module software to meet its needs.

A small academic library that was investigating hosted services utilizing open source software found that the start-up costs for configuration, data migration, and training were slightly higher for the open source solution than two of the proprietary software solutions it investigated, and annual support costs for the open source solution were substantially higher. It is, therefore, essential to compare costs for multiple options, rather than assuming that the open source solution will be less costly.

- *Lack of coordination*

The decentralized development of open source software means that progress can be chaotic and there may be delays in addressing bugs and in completing planned enhancements. This may increase the burden on a library that decides to proceed on its own.

- *Inadequate training technical support*

No training comes with open source products unless a commercial vendor is retained. Documentation tends to be limited and aimed at developers. There usually is limited technical support, especially for users of the software. However, a few open source products do have training and technical support available commercially for a fee.

- *Lack of participation*

Too few participants can cause the development effort to become too expensive for one or a handful of committed parties. Of 34 open source efforts the author has tracked since 2002, only twelve were active in early 2008.

- *Lack of guarantees and remedies*

Unlike turnkey systems using proprietary software, there are no guarantees of quality or performance for open source software. A library may, therefore, find that open source software is not as described or the documentation is deficient. Vendors that provide support services for open source products do offer some guarantees, but no remedies similar to those offered by vendors of proprietary products. Only purchasers of proprietary products can expect financial and other contractual remedies for poor response times and loss of functionality.

- *Scalability and speed*

Open source software may not offer the scalability and speed of proprietary software because the easy-to-use and general-purpose programming languages used are not very scalable and are slower than other languages. For example, the Perl programming language that is widely used in the development of open source software is not very scalable because every script that is run launches separate programs on the server. This means that if one has 100 simultaneous users accessing documents, one could be running 100 to 1,000 extra programs on the server. Furthermore, time is needed to start the Perl program on every script request. Perl is also slow, running as much as 98 percent slower than various versions of C.

Proprietary software usually is written in multiple languages using the FFI (foreign function interface) to combine the speed of the various versions of C for the minority of applications that need it and Java for efficiency for the rest. However, C and Perl can also be a suitable combination.

The following criteria are suggested for evaluating open source integrated library system software:

1. There is current development activity
2. At least the cataloging, circulation, and patron access catalog modules are currently available; and acquisitions and serials control are in development.
3. MARC is supported
4. Current source code and documentation are available for downloading under the GNU General Public License
5. The product is currently in use in libraries
6. Scalability is not an issue—There is no risk of database size or activity levels exceeding the capacity of the software.

In 2005, twelve open source integrated library system products were identified in this TechNote. Development activity apparently ceased for one and one new product emerged by early 2007. Those that appeared to have at least some current development activity underway as of early 2008 were Avanti, Emilda, Evergreen, GNUTeca, Koha, Learning Access ILS, NewGenLib, OpenBiblio, phpMyLibrary, PMB, PYTHEAS, and WEBLIS. Of these, Evergreen and Koha were the most active, and the most widely used.

The following brief descriptions are offered to help a library reduce the number of options it wishes to consider:

Avanti MicroLCS was begun in 1998 as an integrated library system for small libraries of all types, but there was little development activity until 2004. Peter Schlumpf, its developer, appeared to be working on it alone. As of early 2008, cataloging and patron access catalog modules were in general release. A circulation module had been in development for at least two years, but no release date had been set. MARC is supported. The database is limited to 128,000 titles and 256,000 items. The software is written in Java and will run on Windows and Linux. It includes its own database manager. The source code and documentation are not available online because

the developer is not seeking the participation of others. Both open source and commercial (i.e., supported) versions were available as of early 2008. There is a fee for the commercial version. There was an online demo available at that time. No library is known to be using it. While Java would normally make the product moderately scalable, the database management system limits the product to use by small libraries. More information is available at www.avantibrarysystems.com/

Emilda is being developed by CompanyCube (formerly Realnode Ltd), a Finish software company that obtained grant funds to create an open source integrated library system in 2000. The initial system was designed with the assistance of several school libraries. It did not conform to standards and used PHP as the programming language. The current product, which does conform to standards, including MARC and Z39.50, was begun in 2003. It is XML-based and can be run under any operating system. The circulation and patron access catalog modules were introduced in general release on June 29, 2005. No other modules appeared to be in development in early 2008. It uses the Zebra Server from Indexdata as a backend server. The source code and documentation are available online in English. There is also an online demo. The product is available under the GNU General Public License. It was in use at 14 Finish school libraries in early 2008. The product is highly scalable because XML can be ported to virtually any other language. More information is available at www.emilda.org/

Evergreen is an integrated library system for public libraries developed by the Georgia Public Library Service for use by the Georgia Library PINES Program, a consortium of 252 public libraries in that state. The languages are C++ and Perl. The operating system is Linux and the DBMS is PostgreSQL. The server technology is Apache. Work was begun in 2004 and as of early 2007 several public libraries in Georgia were using the system for cataloging, circulation (including offline circulation), and patron access catalog applications. The development of an acquisitions module was begun in 2007 in a collaborative effort between the Georgia Public Library Service and the University of Windsor in Canada. There were also plans to begin development of a serials control module in 2007. As of early 2008, neither of these modules was yet in

general release, but the number of development partners had increased significantly so that general release of these modules was projected for Release 2.0. Releases 1.22 and 1.4 were scheduled to appear in the first half of 2008. The source code and documentation are available online to anyone under the GNU General Public License. The combination of C++ and Perl makes Evergreen highly scalable and, therefore, suitable for large libraries and consortia. Evergreen has been selected by a number of libraries and consortia, including the University of Windsor, Laurentian University, McMaster University, Kent County (MD) Public Library, Grand Rapids Public Library, Michigan Library Consortium, Mayfield Memorial (IN) Public Library, Indiana Open Source ILS Initiative, and British Columbia Pines. Information is available at www.open.ils.org/ Equinox (<http://eslibrary.com/>), a for-profit company, actively markets and supports Evergreen, including a hosting service.

GNUTeca is an integrated library system developed in Brazil for academic and special libraries. Cataloging, circulation, and patron access catalog modules were in general release as of early 2008. The programming languages are Perl and PHP. The operating systems include all versions of Microsoft Windows and Linux. MARC is supported. Source code and documentation are available online in Portuguese, Spanish, and English under the GNU General Public License. Several Brazilian school libraries were using the product as of early 2008. The product is not very scalable because of the use of Perl and PHP, therefore, users who must support more than 50 concurrent users should proceed with caution. More information is available at www.gnuteca.org.br/

Koha (<http://koha.org>) was developed in New Zealand beginning in 2000 by Katipo Communications Ltd. under contract with the Horowhenua Library Trust. LibLime (<http://liblime.com>), a U.S. company that has been providing support for Koha since early 2005, purchased the Koha Division of Katipo Communications in February of 2007. In addition to the development and maintenance work by LibLime, there are volunteers in several other countries contributing to the open source software, and several other companies providing support services. The acquisitions, cataloging, circulation, and patron access catalog modules were in general release as of early 2008. However,

acquisitions and serials control were less complete than the other modules. MARC is supported. The programming language is Perl. Linux is the preferred operating system and the DBMS is MySQL. The source code and documentation are available online under the GNU General Public License. There is also an online demo. More than 200 libraries around the world were believed to be using it as of early 2008, approximately one-third of them in North America, many of them supported by LibLime. The vast majority are small school, special, or public libraries.

Koha is available for free download from the Koha Web site or from one of the companies that supports the open source software. These companies, including LibLime in North America, do not charge for the software, but do charge for consulting, programming, training, technical support, and the hosting services they provide.

The use of Perl as the sole programming language limits Koha's scalability. For that reason, a variant known as Koha ZOOM was developed by LibLime as a commercial version for mid-size and large libraries. It utilizes Index Data's Zebra, an indexing and searching tool. In addition to overcoming the scalability problems in Koha, Zebra includes support for true Boolean search expressions and relevance-ranked free-text queries. The first user was the Nelsonville Public Library in 2006, a library with a collection of 350,000 items in its seven branches and an annual circulation of 600,000. It had previously used Koha, but needed the more robust Koha Zoom.

. The Koha Zoom software will run under either Linux or Windows. It supports not only MARC, but also XML and Z39.50. It also features federated searching. Koha ZOOM Basic consists of installation and documentation media, training materials, and 30 days installation and configuration support. Optional services include data migration, training, software updates, tech and support. Koha ZOOM Appliance comes as a tower or rack-mounted server with pre-installed Linux and applications software, documentation and training materials, and the same optional services as Koha ZOOM Basic. Koha ZOOM Hosted is a fully hosted ASP solution. It includes nightly back-up.

Koha Zoom has attracted a number of mid-size libraries, including the Howard County (MD) Library and the Santa Cruz (CA) Public Library. Consortia that have selected Koha Zoom include ICAN, WALDO, and the MassCat Consortium,

The original Koha is now called Koha Classic to differentiate it from Koha Zoom.

Additional information about the open source Koha product and instructions for downloading the software is available at <http://koha.org/>; additional information about LibLime's support for Koha and its commercial offering Koha ZOOM is available at <http://liblime.com/>

Learning Access ILS is offered by the Learning Access Institute, a non-profit organization in Seattle, to small public libraries. The product, which was originally developed by the Technology Resource Foundation, was previously known as OpenBook. It is loosely based on the original work done in New Zealand by Katipo Communications Ltd. Cataloging, serials cataloging, circulation, and patron access catalog modules were available in general release as of early 2008. Planning for an acquisitions module was announced in 2007, but none was available as of early 2008. MARC is supported, as are Z39.50 and Unicode. The programming languages are Perl and PHP. SQL is the database management system. The preferred operating system is Linux, but the software can be ported to Windows. There is also a turnkey version for Apple MAC Mini known as aVista. There is an online demo, but the product is not available for download. Two small public libraries were using the product as of early 2007. There was no response to an inquiry about additional users in early 2008. The reliance on Perl does not make the product highly scalable. Users who must support more than 50 concurrent users should proceed with caution. The Institute offers project implementation assistance and on-site training. More information is available at www.learningaccess.org/

NewGenLib was developed by Verus Solutions Pvt Ltd and the Kesavian Institute of Information and Knowledge Management in India beginning in 2005. It became an

open source product under the GNU GPL License in January of 2008. The modules include acquisitions, serials management, cataloging, circulation, and a patron access catalog. It uses Java and a back-end database based on open source PostgreSQL. It conforms to MARC 21 and Unicode. It can be installed on Windows or Linux. As of early 2008, more than 120 libraries in Asia had downloaded the software, but there was no information about how many of the libraries were actually using the software. Information is available at <http://www.verussolutions.biz/>

OpenBiblio, which is being developed by a small number of people, has been an off and on again project. Development activity peaked in 2006-2007 with release 0.6.0. There has been no release since then. The product includes cataloging, circulation, and patron access catalog modules. The programming languages are PHP and LAMP, and the operating system is Linux. UNIMARC is supported. There is an online demo and software can be downloaded. There is no reliable scalability information for LAMP. More information is available at <http://obiblio.sourceforge.net/>

PhpMyLibrary began in the Philippines in 2001 as the hobby of a single developer. The target is small academic and special libraries. While the software may be downloaded, the development is highly centralized like Avanti, with the ultimate control of the source code in the hands of the project's founder. Documentation is minimal. There is an online demo. The cataloging, circulation, and patron access catalog modules were in general release as of early 2008. A serials control module was nearing completion. SUSMARC is supported. The operating system is Linux or Windows, and any SQL database system may be used. The programming language is PHP. The software is available under the GNU. Free Documentation License. Scalability is limited. More information is available at <http://sourceforge.net/projects/mylibrary/>

PMB (PhpMyBibli) is an open source ILS based on a product originally developed by a public library in France in 2002. It is now maintained by PMB Services, a French company. The modules available as of early 2008 were acquisitions, cataloging, circulation, patron access catalog, and selective dissemination of information service. The

most recent release as of early 2008 appears to be 3.0, released in the first quarter of 2007. It supports UNIMARC and Z39.50. The product is available in English as well as French, but not all of the documentation has been translated. It was initially available under the GNU General Public License, but it is currently licensed under CECILL, a French equivalent. It is written using the PHP programming language. Scalability is limited. It can be installed on Windows or Linux. A hosted solution is available. Information is available at www.sigb.net/

PYTHEAS was originally developed in 1999 by a librarian at the University of Arizona. After he abandoned the project, a systems librarian at the University of Windsor continued the programming. Only the circulation and patron access catalog modules were available in early 2007. The programming languages are Java and XML. The source code is available for downloading and there is a demo. The product is in the public domain. Documentation is limited. The product is highly scalable. More information is available at <http://web2.uwindsor.ca/library/leddy.people/art/pytheas/> [Note: The Windsor University Library has been working with the Georgia Public Library Service on the development of an open source acquisitions module since early 2007].

WEBLIS is a Web-based integrated library system developed by the Institute for Computer and Information Engineering of Poland with support from UNESCO. It is based on the earlier CDS/ISIS product funded by UNESCO. The cataloging, circulation, and patron access catalog modules were in general release as of early 2008. The programming languages are not identified, but the DBMS is WWW-ISIS. The source code and documentation are available online in English. They are in the public domain. An online demo can be viewed before downloading the software from <http://www.unesco.org/isis/files/weblis.zip/>. The product was in use by a number of special libraries as of early 2008.

Final revised version May 23, 2008