"Open Source" Integrated Library System Software

By Richard W. Boss

A number of public libraries have been investigating "open source" integrated library system software. However, the percentage of libraries that would seriously consider implementing an open source ILS is still small, approximately three percent in a survey of 80 North American libraries conducted by the author in October of 2008. Marshall Breeding of the Vanderbilt University Libraries came up with a similar figure in a much larger international survey in late 2007 (www.librarytechnology.org).

It is difficult to determine how many libraries are actually 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 selected open source ILS software is estimated at more than 500 worldwide, a minority of which have contracted with a commercial company for support services.

The motivation of libraries considering an open source ILS appear to be both financial and a desire to tailor a system to more closely meet their requirements than the proprietary products allow. Interviews conducted by the author in the fourth quarter of 2008 with more than a score of libraries that have chosen an open source ILS found that small libraries focus more on potential cost savings, while large libraries focus on the possibility of tailoring functionality more closely to their needs.

None of the libraries contacted had implemented all of the open source ILS modules that were of interest. Only two had comprehensive cost information. The information herein is, therefore, preliminary and subject to change in the near-term future.

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 both open source and proprietary products that are based on open source software.

While the software may be free, a developer or distributor may charge for services,

including special programming, configuration and installation, training, file migration, technical support, and hosting services for libraries that do not want to implement and maintain an in-house system. Some companies that distribute and support open source software are for-profit organizations and may charge prices for their services comparable to those charged by proprietary software vendors.

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.
- *Perceived Low cost*

  There is no charge for the software itself; therefore, the capital outlay required by proprietary software is avoided. The major software 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 costs*

  Open source software tends to be less complete than proprietary software. Acquisitions, serials control, and interlibrary loan modules may be missing or incomplete. Online ordering and claiming, self-check, email/telephone/SMS patron notification, e-commerce, RFID support, I&R (information and referral), OAI-compliant harvesting, and program registration are generally not available. A library may find that it needs to do a great deal more work than anticipated to adapt the software to local needs or to add modules or sub-modules that are not a priority for other users of the open source product. A library may not realize how much work will have to be done to provide it with the breadth of functionality that is found in proprietary products. When it does decide that more development is needed, it must retain staff, contract with a company that provides that service, or seek to solicit participation in development from the open source community.

Documentation and training may also add more cost than expected.

- *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. There is no effective way to pressure libraries that have committed to development work to stay on schedule. This may increase the burden on a library that decides to proceed on its own.

- *Inadequate training, documentation and technical support*

    No training comes with open source products unless a company that provides that service is retained. Documentation tends to be limited and aimed at developers. User documentation usually must be developed locally. There is limited technical support. However, a few open source products do have training and technical support available commercially for a fee.

- *Lack of participation*

    Too few participants involved in development 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 and only two of those—Koha and Evergreen— had software development, configuration and installation, file migration, training, and trouble desk support available from commercial companies.

- *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. Companies 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 with at least two substantial releases a year.
2. At least the cataloging, circulation, and patron access catalog modules are currently available; and acquisitions and serials control are in development. If a library requires broader functionality, there are similar libraries that may share development priorities.
3. MARC is supported
4. Current source code and technical documentation are available for downloading under the GNU General Public License
5. The product is currently in use in a significant number of 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, the author identified twelve open source integrated library system products. 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. However, only Evergreen met all of the foregoing criteria and had been implemented by more than a score of libraries. Koha did not meet the scalability requirement, but Koha Zoom did and had been implemented by almost a score of libraries.

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 late 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.avantilibrarysystems.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 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/.

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 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/

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 Perl and, to a lesser extent, C. The operating system is Linux and the DBMS is PostgreSQL. The server technology is Apache. The staff client user interface is written in Mozilla XUL (XML and JavaScript). The patron access catalog can be accessed by any Web browser.

Work was begun in 2004 and by 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 late 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 in 2009. 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's SITKA. Not all had gone live by late 2008. Information about Evergreen is available at www.open-ils.org/. Equinox (http://www.esilibrary.com), a for-profit company launched by several of the people who worked on the Georgia Pines project in 2007, actively markets and supports Evergreen, including software development, configuration and installation, file migration, training, trouble desk support, and hosting service.

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. LibLime is wholly owned by the holding company MetaVora, Inc., which is wholly owned by the four principals in LibLime. 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 substantially 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, file migration, 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 programming, data migration, training, trouble desk 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 to large 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 mid 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 late 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 mid 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 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

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 mid 2008 were acquisitions, cataloging, circulation, patron access catalog, and selective dissemination of information service. The most recent release as of mid 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 cquisitions module since early 2007, but no release had been issued as of late 2008].

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 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 mid 2008.

Cost Elements

Regardless of whether it is motivated by a desire to save money or to have an integrated library system that meets all of its needs, a library should seek to ascertain the total cost of an integrated library system before making its decision. Only then can it know whether open source system is less expensive than a proprietary one or how much it should budget for a fully implemented system that has the potential to meet all of its requirements.

The costs of proprietary ILS products are relatively easy to ascertain because most are purchased using an RFP (Request for Proposals) and competitive bidding, and there are many libraries with experience that can be contacted for information. It is more difficult to ascertain the cost of open source products as competitive bidding using an RFP is rarely

undertaken, and few libraries have fully implemented an open source system. Nevertheless, it is possible for a library to get some idea of the cost of an open source ILS if a checklist of cost elements is used. The following are the key elements:

Software License

The most obvious cost saving realized with open source is the absence of software license fees, fees that can range from as little as $30,000 for a very small multi-function integrated library system to $500,000 or more for a very large one.

Software Development

A small library that requires only the bibliographic maintenance, circulation, and online patron access catalog modules may not need to undertake software development in-house or by purchase from a company that provides that service. However, a library that requires a broad range of functionality may have to invest in software development. Large, complex libraries may find that the available open source software has been written for libraries quite unlike them. Extensive software development to enhance an open source ILS can cost from $100,000 to $250,000 per module.

Central Site Hardware

While hardware costs for mounting an open source ILS are comparable to those for a proprietary product, they are less predictable for large and very large libraries because almost all of the experience with open source software has been with small to mid-size libraries. A very crude rule of thumb is to budget $300 per concurrent user the system is expected to support for central site hardware. In either case, client and network hardware may be as much as $2,500 per client.

File migration

File migration is one of the most widely purchased services by libraries implementing an open source system. The price typically is $.15 per record migrated for small libraries, approximately 50 percent higher than vendors of proprietary systems charge. Mid-size and large libraries have generally negotiated lower unit prices for migration of records to an open source system, thus making them more comparable to those offered by vendors of proprietary systems.

Configuration and Installation

These services may include profiling, hardware installation, software installation, file migration, and system testing. The minimum for an open source system appears to be $25,000; for a proprietary system, $18,000.

Documentation

While technical documentation for open source software appears to be usable with

minimal local expenditure of time unless local development is undertaken, user documentation has required considerable local effort. Several libraries have reported that each module requires at least 160 hours of time. Assuming a cost of $40 an hour, including salary, benefits, and overhead, the total would be $6,400 per module. User documentation by vendors of proprietary systems is much more complete and not separately priced. It usually requires only online editing to localize it.

Training

With the exception of very small libraries that purchase training for the entire staff, most training is done in-house by a small number of trainers who have been trained by the vendor of a proprietary system or a company that provides such service for open source software.

The general rule of thumb is a minimum of two days of training for a system manager and two days of on site training in each module for system users. The minimum number of days of training is eight for a library that implements only bibliographic maintenance, circulation, and online patron access catalog. For a comprehensive system, it remains two days of training for a system manager, but the number of days of training for system users increases to twelve. Vendors of proprietary software generally charge $1,500 per day for on site training. The on site training offered by companies that support open source software is as high as $2,800 a day. While Web-based training is available for both proprietary and open source systems at $125 to $150 per day, libraries report that it is much less effective for initial system training than on site training by a specialist.

Ongoing support

The ongoing support services available for both proprietary and open source software are trouble desk, bug fixes, and software enhancements. Vendors of proprietary software generally bundle all three at an annual cost of 15 percent of the undiscounted purchase price of the system. The first year is almost always included in the initial purchase price. Open source support uses tiered pricing that is not directly related to the purchase price of the system. The minimum appears to be $9,000 to $15,000 per year beginning at installation, with the cost depending on the number of hours of trouble desk support. Bug fixes are included, as are enhancements made by participating libraries. However, bug fixes and enhancements undertaken under contract with a library are billed at a minimum of $150 per hour.

A library that undertakes in-house development will need to maintain the software it has developed. The cost is at least 20 percent per year of the initial development cost.

Case Studies

The following case studies are based on telephone interviews and review of documents supplied by several libraries. nfortunately, only Libraries A and B had full cost information. The other libraries either lacked information or were unwilling to supply it even though

confidentiality was committed.

Library A consists of a small main library and three very small branches. There are 25 full-time staff. Members. The library has been automated for more than five years using a proprietary integrated library system. The modules that it has used on that system are serials control, cataloging, circulation, inventorying, and patron access catalog. It issued an RFP in 2007 that specified all of the foregoing modules plus acquisitions. Distribution was limited to five vendors of proprietary integrated library systems. The lowest bid that met all of the requirements in the RFP was $51,540. The annual cost after year one was $7,200. The vendor was willing to commit a five percent per year cap on increases. The proposal included central site hardware and the migration of bibliographic, patron, and transaction records. Authority control processing was also included. The vendor offered eight days of on-site training. The total amount payable to the vendor over a five-year period was estimated at $82,573. The existing staff would not have to be increased.

The library then decided to pursue an open source ILS solution.

A for-profit company that supports open source software quoted $45,000 for central site hardware and set-up, plus $9,000 per year in annual hardware/software support for Monday-Friday 9:00 to 5:00 and $15,000 for 24/7 beginning at installation. A five percent cap on annual increases was committed. The acquisitions and serials control modules were not yet complete. The library was quoted a 25 percent premium for participation in development. No on site training was included and migration, which was separately priced at $6,000, was limited to bibliographic records. Web-based training was quoted at $125 per hour or $8,000 for eight full days. The library was later offered on-site training at $2,800 per day. Assuming that eight days of on-site training and the lower priced support option were selected, the total amount payable to the vendor over five-years was estimated at $129,563. While the library would not undertake in-house development, it would have to retain a consultant to assist in the migration of serials and circulation transaction records at an estimated cost of $5,000.

The foregoing suggests that a small library that wants broad functionality and extensive support somewhat similar to that which a vendor of a proprietary system would provide would pay more for open source.

Library B is a small library that has been using a proprietary integrated library system for more than six years. It has used only the bibliographic maintenance, circulation and patron access catalog modules. It planned only to use these modules of an open source product. The library decided to pursue an open source ILS and was quoted $62,000 for hardware, set-up, data migration, and four days of on site training, plus $9,000 per year for ongoing support beginning in year one. A cap of five percent per year in subsequent increases was submitted. The five-year cost was estimated at $111,731. The library had estimated that the five-year cost of changing to a new proprietary ILS would be $132,500. The library decided to pursue the open source option.

The foregoing suggests that a small library that requires only bibliographic maintenance, circulation, and patron access catalog may realize lower cost with an open source product.

Library C is a mid-size library with several branches, a collection of more than a million items and a staff of more than 150 people. It had shared a proprietary integrated library system with a number of small libraries in a regional consortium. It began to investigate other options after becoming frustrated with the inability to maintain procedures different from those favored by the membership as a whole. It was also concerned about the cost of the proprietary system. It, therefore, decided to investigate open source ILS software. It already had a systems staff of four and thought that it would only need to add a half-time system operator to have its system in-house. That suggests that the existing systems staff would have to be involved in system operation as a halftime person could only cover a small percentage of the hours the system would be operational each week. However, no time or cost had been estimated as of late 2008.

The library subsequently entered into an agreement with another consortium for a joint implementation of an open source shared ILS that would be based at Library C. The library handled its own hardware and software implementation, but contracted with a commercial company that provides support for the open source ILS for data migration, minimal training of the consortium's trainers, and a support desk. The consortium added one full-time person to write user documentation and shuffled the duties of two others to commit part of their time to the training program. It charges members that wish to participate a fee based on a formula that uses collection size and circulation activity. It contracts with Library C for the operation of the central site. That partially offsets the cost of the library's participation in the consortium's integrated library system program. As the majority of libraries that expressed an interest in participating had not yet gone online, the consortium had no hard cost figures as of late 2008.

Detailed cost information for Library C was also not available in late 2008. The library estimated that it would save almost $90,000 by leaving the shared proprietary system—an amount payable for membership and automation support services by the old consortium and the vendor of the proprietary system. However, it had not calculated the cost of in-house staff reassigned or hired to manage an in-house system. The staff costs attributable to the open source system by the library may be at least $60,000 a year, and that by the consortium at least $25,000. As only the circulation and patron access catalog modules had gone live by late 2008, and those for only a few weeks, it was not possible to ascertain what the staffing requirements might be after full implementation. The library had a commercial vendor of open source integrated library systems handle migration of the database at a negotiated confidential price. It did not yet have figures on the amount it would pay to participate in the new consortium's automation program, nor how much it might receive from the new consortium for managing the central site for the system. Finally, it was not yet possible to ascertain whether the planned acquisitions and serials modules would meet the library's needs without local development work. The library had already built its own telephone renewal system using existing staff.

It may be mid to late 2009 before it is possible to determine what the costs of implementing an open source integrated library system in a mid-size library system.

Library D is a large library with several branches. Circulation is in excess of five million a year. While it has had a proprietary integrated library system for several years, it has had extensive experience with open source software, including the Abunta operating system

for desktops, Firefox as a browser, and OpenOffice. The library has a systems staff of six. It pursued an open source integrated library system to replace its proprietary system less to reduce cost than to increase control over development. It downloaded the open source software onto one of the three servers that it had been using
for the proprietary system. It contracted with a commercial company that supports the open source product selected for three days of onsite training and data migration. The amount was negotiated and confidential. The library has submitted a request for quotation to the commercial company that supports the open source system for a series of enhancements. Based on a partial response, the amount is expected to be in excess of $100,000, primarily for a robust acquisitions module. No serials module was contemplated as of late 2008. The library will also commit one staff member to in-house programming and another to the writing of user documentation and training. That might cost in excess of $100,000 a year.

The open source system is scheduled to go live in the second quarter of 2009, therefore, no detailed cost figures were available in late 2008. However, the library administration has expressed the view that it is committed to using open source software whenever it is appropriate. The criteria are the flexibility to make the interface between the integrated library system and the library's Web site transparent and to be able to determine enhancement priorities rather than relying on the choices made by a vendor of proprietary system. At the time of the interviews, there was no expectation of cost savings.

The tentative conclusion is that a large library may not save money with open source, but may be able to realize functionality tailored to its unique needs.

Library E is a very large multi-branch library with an annual circulation of more than 15 million. It has had its current proprietary system for more than five years. It has been investigating open source ILS in the hope of improving staff efficiency, especially in circulation, and expanding functionality in all modules at proportional cost to the replacement of its current system when it is seven years old. While it has a systems staff of almost 20 people, it does not plan to undertake in-house software development. It has obtained a quotation from a commercial company that supports an open source ILS to rewrite the existing open source circulation module to meet all of the needs of the library. The quotation was for more than $220,000, plus an estimated $50,000 for hardware for supporting the circulation software. No documentation, training, or migration services were quoted. The library will evaluate the resulting circulation product in mid-2009. If the evaluation is positive, the library expects to go live on circulation and other open source ILS modules the following year. The library
administration and head of systems remain open minded about open source. They will proceed only if any higher cost for an open source solution delivers additional functionality that is worth the difference.

The tentative conclusion is that a large library that requires considerable software development to meet its needs will not save money with an open source solution, but it may have a system that is tailored to its unique requirements.

Revisions completed December 14, 2008