TO: PINES Member Libraries
FROM: David W. Singleton, Deputy State Librarian
DATE: June 10, 2004
SUBJECT: PINES SOFTWARE DEVELOPMENT

Following last week's announcement by Dr. Veatch that PINES would be pursuing in-house development of a new integrated library system (ILS), I know that many people may be asking, "Why would PINES develop its own ILS when there are vendor-developed systems on the market?" I wanted to take a few minutes to give everyone some background on how and why this decision was made.

When many librarians think about homegrown ILS systems, they likely recall images of ugly, clunky, and buggy software that couldn't get any more user-unfriendly. Many of these older in-house development systems were in existence in the 1980s and even into the early 1990s. During the past 20 years, almost all of these in-house development systems were migrated to vendor-developed systems.

How is the PINES ILS going to be different from past models? And how has software development changed over the past 20, or even 10, years, to make this possible?

You may find it interesting to learn that programmers refer to collections of building block software as "libraries". Like librarians, programmers like to improve their libraries, but unlike librarians, programmers never have to throw away, or "weed", anything. When they do "weed" material, it's more like taking the best parts from old books and writing a new and better book with that material. With the advent of the Internet, high-speed telecommunications, and cheap personal computers and storage space, programmers also have an advantage of being better able to "lend" programming material and share it with others around the world.

To get an idea of just how much software development has changed during the past 10 years, think of the simple example of an old library card catalog versus a computerized library catalog. Although I love the look and feel of the old library catalog, the fact is that electronic data and databases have made searching much more flexible than just the old standards of author, title, and subject. The evolution of software libraries has been at least as dramatic. For example, in the mid-80's, the programming language called Basic was more or less free (it helped to sell hardware), and contained a few nifty commands for drawing lines, dots, and circles on a computer monitor in various colors. Programmers could also make music by entering sound commands specifying the frequency and duration of a note. For a few hundred dollars, programmers could buy a C or Pascal compiler and a special software library or two for drawing even prettier and faster 2-dimensional pictures or for making a synthesized voice come out of the PC speaker.

These days, there are ubiquitous "commodity" libraries and compilers that programmers can get completely for free; these are capable of modeling real-time 3-dimensional graphics and fielding high quality multi-channel sound samples. Even better, because these tools are open source, programmers have access to information such as how these tools were designed and written. This allows programmers not only to learn from open source software but also to contribute to its evolution. None of these things are easy for a single person to create from scratch, but in a worldwide atmosphere of collaboration and sharing, they come together quickly.

When Jason first started working with the MARC data format, he did not have to learn the low-level bits and bytes of how to read the raw data. This was because another group of programmers had already done the hard part for him by developing a MARC software library, and then freely released the code under an open source license. So, instead of having to write a thousand lines of tedious computer code, Jason has been using these MARC libraries in de-duplicating our bibliographic data with a hundred lines of code. Because he has control over this code, he can modify the program as necessary to be specific to the needs of PINES.

In developing the new PINES ILS, we are fortunate in that many of the building blocks have already been completed for us, and are waiting to be utilized. This was not true a mere 5 years ago. For example, instead of spending countless hours attempting to read and write native MARC data, we can utilize the open source MARC software library. This is true for many functional areas of the ILS, and this greatly facilitates developing a modern and sophisticated software package, such as an ILS.

The major advantage of developing our own ILS is that we have complete control over the functionality of the system, and can truly tailor the ILS to the very specific needs and wants of the PINES community (public and staff). This cannot be said for any vendor-developed product currently on the marketplace.

Who is involved in the PINES development project?

Brad LaJeunesse and Jason Etheridge will lead a team of programmers in the software development effort. We plan to hire three additional programmers with specialties in database design, GUI and interface design, and systems-level programming. This development team will work closely with the existing PINES staff, utilizing the knowledge, skills and experiences of each member of the team in creating a system that addresses the unique needs of PINES.

The design team will be located in the new OIIT facility in Oconee County, just outside Athens. This proximity to OIIT staff will enable PINES to take advantage of the wide range of expertise found among the BOR staff.

Regardless of whether PINES staff members are located in Atlanta or at the new OIIT facility, all PINES staff will remain under the supervision of the PINES Program Director, Julie Walker. I will be working with Ms. Walker to ensure that the PINES program has the staff and technical support needed to both operate the current system at optimal levels and to design and develop the software needed for the future of PINES.

How will PINES member libraries have input into the development process?

There will be many opportunities for the PINES staff to gather ideas, suggestions, and specific requests from the PINES community. We plan to convene a series of focus groups around the state, where ideas can be discussed and refined. These meetings will be scheduled and geographically located so that all interested PINES library staff can attend.

The standing PINES subcommittees, with their specialized knowledge of the various functional modules, will work with the development team at every step to create a system that is easy to use, intuitive, and workable for PINES libraries.

The new PINES Development Committee, newly formed by the PINES Executive Committee, will serve as the clearinghouse for ideas, assisting staff in determining direction and focus. Web forums, such as the current forum on reporting, will be valuable tools for PINES members to discuss issues as they arise. Helpdesk tickets will continue to be a source for developers to identify issues and problems that need to be addressed.

As always, the PINES team will be available to listen and respond to questions and ideas; the number one reason for developing our own software is to create a solution that most closely fits the needs of PINES. The input of every PINES library is valuable in identifying those needs and determining how to best address them.

Why were vendor-based solutions not right for PINES?

The PINES staff has spent significant time and effort in exploring every ILS product on the market today, utilizing demonstrations, conversations with other consortial customers, and a vast quantity of published materials on the library automation marketplace today. It became rapidly apparent that only three products were appropriate for PINES in terms of platform and scalability. Our current vendor is one of those. One of the three stated that they could only handle PINES through dividing the database, not utilizing a single system as we have done for 4.5 years; we felt that this was a step backwards that we were not willing to take.

An acquisitions module for a large consortium has proven to be a show-stopper for every vendor-based solution. Our contacts in the consortia community confirm that no available system offers a workable acquisitions product, something our libraries require.

Each system that we explored had strengths and weaknesses, but no system was designed for a consortium with the volume and mix of transactions that PINES performs each day, and any off-the-shelf product would require significant modifications in order to function adequately in our situation, if that modification was even allowed under the vendor's license agreement.

In discussing in-house development of PINES software (as opposed to vendor solutions), a branch manager in a PINES library recently commented: "You know, I think it's even more of a risk for us to go to all the trouble of moving to another product, and then being just as unhappy as we are now."

Why are we certain that this is the right direction for PINES software development?

There are a few basic questions that I've asked Brad, Jason, Julie, and the PINES staff to answer. These are questions that I asked many months ago, as we were learning about the possible options for PINES. This may be more than you want to know, but we wanted to give you some background on open source and why this direction was chosen for PINES.

Exactly what is "open source," and how do open source software licenses work? Open source software is also known as free software. The term "open source" refers to several things:

1. A specific set of software licenses.
2. A practical method of developing software.
3. A community of developers and users who support #1 and #2.

A free (open source) software license must, among other things:

- allow others to freely redistribute the software
- require that the source code for the software be available
- allow derived works based off of the software
- not discriminate against persons or groups
- not discriminate against fields of endeavor
- be applicable to all who receive the software

Open source (free) software licenses promote the sharing of software and the code for creating software. This in turn promotes the formation of diverse communities, which collaborate on improving free software.

Linux, Apache, and MySQL are examples of open source software that is currently available and being used extensively worldwide. Linux is a popular open source operating system (comparable to Windows) developed by a worldwide team of volunteer developers. There are many Linux websites, but http://www.li.org has excellent background on Linux and how it is used.

Apache, another open source software package, is the most popular webserver software on the internet. Webserver software like Apache is what allows computers to “serve” webpages. For example, when you point your browser to http://www.amazon.com, the server you are connecting to is utilizing Apache to send your computer the webpages. Currently, approximately 67% of webservers on the Internet utilize Apache. The next closest competitor, Microsoft’s proprietary IIS runs on about 21% of the internet’s webservers. Apache, like Linux, is developed as open source software by a worldwide team of volunteers.

And MySQL, the open source database upon which the PINES reporting system is based, is the “world’s most popular open source database” (http://www.mysql.com). MySQL offers much more flexibility and speed in the PINES web-based reporting system than the existing (old) reporting system.

We decided on open source for our development model for both pragmatic and philosophical reasons. The open source community is a natural ally of the library community. Both try to enrich their members through sharing and disseminating knowledge, and both are open to everyone, private or public, commercial or non-commercial.

The practical benefits are extraordinary. In successful cases like Linux and Apache, the number of people who help out (whether they write code, or documentation, or test the system, or merely offer suggestions) can outnumber by far the manpower a conventional software house can muster. Open source tools and components such as the GNU C Compiler and the MySQL database have also benefited from a "thousand eyes," and these projects are in turn used as infrastructure for creating yet more free software.

By having access to such tools, we can rapidly develop programs and structures more quickly and efficiently than was possible even five years ago. Software components are quickly becoming commodities, and open standards are allowing these components to work together better than before. We no longer have to reinvent each wheel. Now, a programmer can write software in a day that in that past would have taken weeks to develop.

There is already much activity in the open source community revolving around libraries. Koha is an open source ILS that is being developed in New Zealand. However, many other libraries throughout the world have adopted it, including a public library in Ohio. http://www.koha.org. There are also numerous open source tools for working with open standards like MARC, SIP, and NCIP. And there is a lot of interest in the things that PINES has been doing, such as the reporting system and Standalone, both of which were developed in-house.

What about the future, long-term development and support of an open-source system?

One of the nice things about having the freedom to modify our own source code is that we will be no longer constrained by the limitations of current software. We eliminate the risk of something called "vendor-lock, in which the capabilities of the system are limited by the functionality of the software, with no room for modification. But we also eliminate "Brad-lock" and "Jason-lock," in that our open system will be owned by GPLS and we will be free to hire other developers and system administrators to further develop or maintain the system if required.

The software itself will be designed for maintainability. By using modern development techniques and open tools and standards, the whole system will be "transparent" and obvious to any skilled developer. One of the major points of open source software is that you're supposed to share it, but this is of no use if the software is some arcane impenetrable black box. For this reason, open source software typically evolves into something that is continually easier and easier to work with.

By fostering a worldwide community around our software, we're actually creating outside expertise in the system, and safeguarding our future.

What about international standards for library software?

Many of you may also be wondering about standards, and how an open-source ILS can conform to the many library standards including SIP, NCIP, etc. The developed ILS will conform to all library standards, where applicable. For example, we will be MARC 21 compliant, we will have a SIP2 interface (think self-check, print and computer management software, etc.), NCIP interface (to interface with other systems such as GIL and other public library systems), and we will work with vendors on electronic acquisitions ordering. We will also conform to non-library standards such as SQL.

Who will be running PINES during the development process?

You may be wondering who will be responsible for running PINES during the development of the new system, since over the next two years, Brad and Jason are going to be primarily involved in the software development effort. Other members of the PINES team will take additional day-to-day responsibilities from Brad and Jason. Cross-training already is important within the PINES team, so other staff members know how to handle typical situations or problems. Katherine Gregory, a skilled and experienced system administrator, will take the lead in working with the Unicorn system. Julie Walker also has extensive experience in managing integrated library systems.

Our goal is to maintain Unicorn as a stable, reliable system, to continue to offer a high level of service to our libraries, and to continue our work to refine and clean up our database with the goal of customer service satisfaction throughout PINES. While development efforts will be on-going, critical PINES problems will, of course, get priority attention, and immediate needs will come before software development duties.

What are our plans for PINES hardware?

Another critical question is related to hardware for PINES. What changes are needed, and how will these be accomplished? There are two equally important parts to this question. First, the PINES server is aging rapidly. As announced at the PINES Annual Meeting, we are looking at replacing the hardware for the PINES server later this year. This is not as a result of the ILS decision; the PINES server is 5 years old, and is at the end of its operational lifetime, and it needs to be replaced in any scenario. Upgrading not only gives us a more reliable and faster server, but also decreases the annual support costs incurred by our Sun support contract.

Secondly, there is the issue of the workstations currently installed in the libraries. Some of the workstations have been around since 1999 (or even before), and need to replaced regardless of ILS changes. There are two kinds of workstations involved here. First, the OPAC workstations: any workstation that can run a web browser will be acceptable for an OPAC workstation. (Note this includes any OPAC workstation you have running now.) Secondly, there are the staff workstations. Because staff workstations need to do much more than an OPAC workstation, these computers will need more "horsepower" to run. However, it will be one of our goals to make the staff client as "thin" as possible, as we will want the staff client to be able to run on as many computers as possible in your library. While I cannot give exact minimum system specifications before the staff client is even developed, I am very comfortable in saying that the staff client will most likely not work well (if at all) on the computers from 1999. By the time we roll out the staff client, these computers will be 6 years old, and well past the end of their service lifetime. However, I would be comfortable in saying that a computer with at least a Pentium 3 processor and 128 MB of RAM would be fine as a staff workstation.

The 2004 Bill and Melinda Gates Foundation grant, along with significant LSTA (federal) funds, will help to purchase computers to replace at least one-third of all public access computers in Georgia public libraries. This should be in early Fall 2004. We will send out more details as we have them.

How long will the development of a new system take?

The total project, from start to finish, will take approximately 2 years. Note the schedule below is a rough estimate. Each of these categories is not meant to be a watertight compartment, and often the development team will be working on more than one of these areas at a time, especially when two or more of the areas converge on a single development point. With those caveats, here is a rough breakdown of the two-year development period:

Estimated timeline:

- 2 months: Hiring, Planning (focus groups, etc), development of blueprint.
- 3 months: DB design, underlying infrastructure development, prototype clients/interfaces.
- 3 months: "Cataloging": titles, copies, MARC compliance, authority, database searching.
- 3 months: "Circulation": Basic check-in, checkout, holds, patron information, bills.
- **Alpha Release**
- 2 months: OPAC.
- 2 months: Acquisitions
- 1 month: serials
- 2 months: wrap-up, cleanup, packaging of staff client, prepare for beta release.
- **Beta Release**
- 6 months: fixes, further enhancements, training, more training, more training, more training, preparing for implementation.
- **Final release** (approximately May 2006)

Also note at the "go live" (final release), we will only bring up critical areas of the software for production such as basic cataloging and circulation. Other functionality will be brought into production in the order of their importance to daily library operations. For example, it is conceivable that we will not bring the serials module into the production environment until as late as 3 months after the go live date. This staggered roll out of the system will give us the best chances for a smooth and successful implementation.

I know that many of you will have additional questions that are not answered by the above information.   We will be sharing additional plans and asking for your input, which will be critical to the success of this project. If you have questions or concerns, please don’t hesitate to contact Julie Walker (jwalker@georgialibraries.org) or myself (dsingleton@georgialibraries.org).