Case Study: Nepal Evergreen ILS pilot

Project overview

The eIFL-FOSS ILS project provides an opportunity for evaluation of and possible migration to a free and open source software (FOSS) integrated library system (ILS). It promotes direct engagement with the FOSS communities of a chosen ILS. And it lays the groundwork for a mutual support network across eIFL.net for further libraries investigating migration to a FOSS ILS. The project focused on evaluating two FOSS ILSs with robust and open development and support communities: Koha (http://www.koha.org/) and Evergreen (http://www.open-ils.org/). Following a general call for participation in early 2008, libraries from 12 countries representing 11 language groups submitted applications to become pilot sites. From these, 7 sites were selected that represented a wide range of libraries and regions. All applicants joined an email discussion list in order to follow the progress of the pilot sites. The 7 pilot sites are: Fundamental Scientific Library of the National Academy of Sciences, **Armenia**; National Scientific Library, **Georgia**; Mzuzu University, **Malawi**; Library of the Faculty of Medicine Pharmacy and Dentistry, University of Bamako, **Mali**; Madan Puraskar Pustakalaya, **Nepal**; Midlands State University Library, **Zimbabwe**; and An-Najah National University Library, **Palestine** (West Bank). Each pilot library participating in the project was invited to participate in an intensive technical training workshop in Yerevan, Armenia, in June 2008. Case studies of each pilot draw out learning points for other libraries seeking to move to a FOSS ILS.

Pilot library

Madan Puraskar Pustakalaya (MPP http://madanpuraskar.org/) is a non-profit, non-governmental library. It maintains an archive of books, periodicals, ephemera, and other collections in the Nepali language. The collection grew out of the personal collection of Nepali literary materials of MPP's founder and chairman Kamal Mani Dixit. As the collection grew larger it began to attract gifts and donations of further materials from literary personalities, statesmen, scholars, institutions, and from other countries as well. In 1957 Madan Puraskar Pustakalaya was established as a library from this personal collection. It currently holds 26,896 monographs, 5,136 periodicals, and several collections of newsletters, posters, pamphlets, banners, calendars, manuscripts, reports, manifestos, letters, sketches, photographs, negatives, film-footage, speeches, press releases, images of wall paintings, invitations, audio-records, etc. - a unique and rare collection of ephemera in the Nepali language. The main objectives of the library are to collect, preserve, catalogue, and provide access to the library materials for researchers and scholars. Cataloguing of monographs and periodicals, and preservation of rare and endangered materials using microfilming and digitization, are regular activities of the library.

Nepal is one of the newest members of eIFL.net. It has a rich history and a strong library community. Its challenges lie with its infrastructure. Load shedding, or rolling blackouts, means that homes, businesses, and libraries in Kathmandu have electricity for only 8 hours per day. Despite these challenges some inspiring work is taking place in the libraries. MPP, for example, has led the work on Nepali Unicode as well as NepaLinux. Although the technical team at MPP has varied over time there is a wealth of experience there and a willingness to investigate new opportunities.

Motivation for exploring a FOSS ILS

MPP currently uses a cataloguing system for its library developed in-house. The system has a PHP front-end for its web user-interface or OPAC and its cataloguing interface, and a MySQL database back-end to store its data. It is a simple solution that met the needs of MPP initially. However, the data is not stored in one of the internationally standardized formats such as MARC21. This makes the current solution vulnerable. It will always depend on in-house support and development. Moreover it cannot take advantage of the substantial feature set available in modern integrated library systems (ILS), FOSS as well as proprietary. A move to a well-supported FOSS ILS with a comprehensive set of modules would create the consequent need to migrate MPP's catalogue data to a more standard format. Indeed this data-migration itself might be thought of as the key aim in such an effort. It would also open up new possibilities, e.g. interoperability with other standards-based library software, and the flexibility to choose amongst multiple ILSs that support MARC formats. Of course moving to a well-used and community-supported FOSS ILS also introduces the potential advantage of removing the overhead of maintaining and further developing an in-house solution.

> ***Learning point: migration from one library system to another is not an end in itself; it always serves the larger aims of the library.***

FOSS ILS choice

MPP opted to pilot the Evergreen ILS (http://www.open-ils.org/). Evergreen is an enterprise-class library automation system that helps library patrons find library materials, and helps libraries manage, catalogue, and circulate those materials, no matter how large or complex the libraries. Evergreen supports MARC21 format for its catalogue with a PostgreSQL database as its back-end. It is typically deployed on Debian-based servers, though other Linux distributions may also be used. Dibyendra Hyoju carried out the installation and testing under the direction of MPP Director Amar Gurung.

Initial installation

To launch the eIFL-FOSS ILS project a training workshop was held in Yerevan, Armenia, in June 2008 (http://www.eifl.net/cps/sections/services/eifl-foss/ils/ils-project-workshop). Unfortunately, Dibyendra was unable to join the other technical leads of the pilot sites for the training event. As a result he did not get the direct benefit of Dan Scott’s tutelage. Dan is the Systems Librarian at Laurentian University in Canada. He is also a significant contributor to the ongoing development of Evergreen. His intimate knowledge of both the software and the way in which to get useful support from the Evergreen community made his training workshop immensely valuable. Nevertheless, as in most cases, we make do with the resources immediately available to us. Dibyendra established regular email contact with Dan (even whilst we were in on location in Yerevan) and Dan has been a touchstone throughout his initial Evergreen testing.

The first trial installation at MPP was with Evergreen 1.2.2.0, the current stable release at that time. It was installed on a Pentium 4, 2.26 Ghz PC with 512 MB of RAM running the Debian 4.0 GNU/Linux operating system (known as Debian ‘Etch’). This version of Evergreen requires OpenSRF 0.9 and the corresponding staff client that matches the Evergreen version number 1.2.2.0.

> *Operating system: Debian 4.0*
> *Hardware: Pentium 4, 2.26 Ghz PC, 512 MB RAM*
> *Evergreen version: 1.2.2.0*
> *OpenSRF version: 0.9*
> *Evergreen staff client: 1.2.2.0*

According to the Evergreen documentation wiki, the above hardware should be sufficient to install Evergreen for testing purposes. Of course actual system requirements for a live service will depend on how large the catalogue would be and the probable number of simultaneous users of the system[1]. The Evergreen documentation is less clear about the specific skills that would be required for installation assuming, perhaps fairly, that Linux systems administration skills at more than a basic level would be a prerequisite. The dependence of Evergreen on a PostgreSQL database implies that the ability to manipulate such a database will be a prerequisite for use of Evergreen beyond mere testing.

Evergreen in part consists of a large number of PERL modules. These must be loaded on to the destination server in the appropriate order. While this in itself is not problematic for anyone with Linux systems administration skills, the modules do take considerable time to download especially if one has relatively low bandwidth. That requires a reliable Internet connection as well as other infrastructure we sometimes take for granted. In Nepal, at the moment, there is regular load shedding in the electrical system, which means that one is significantly limited in the number of hours that electricity is available each day. That makes installations that require lengthy downloads (involving several hours) from the Internet challenging. This situation is not unique to Nepal.

> ***Learning point: almost any large-scale systems migration will involve significant demands being placed on the available technical infrastructure.***

Setting up the various configuration files for Evergreen, e.g. opensrf_core.xml, opensrf.xml, and ejabber.cfg, requires more than basic Linux systems administration skills. Care especially needs to be taken in following the installation instructions and some immediate and direct support can make all the difference. There are almost always puzzling error messages that show up during an initial installation of such complex software. Knowing what they mean, or better, knowing who to ask what they mean can relieve a great deal of anxiety.

This raises two key components of any FOSS ILS installation: documentation and support.

Documentation

Documentation is not always perfect in FOSS projects. Indeed technical documentation for any software is rarely fully satisfactory. The writing of technical documentation such as installation documentation requires substantial knowledge of the software. Therefore it is very often the developers themselves in a FOSS project who are called upon to write such documentation. Of course the software changes quickly in development. And this means that developers sometimes prefer to hold off writing the technical documentation until the code has stabilized as it does near the point of a version release. Mature FOSS projects with substantial user and development communities make documentation a priority. Documentation writing simply becomes part of the development cycle. The Evergreen project is no exception. It takes documentation seriously as can seen in the vigorous discussion of documentation issues on the Evergreen project site’s dedicated documentation discussion email list.

However documentation, just like source code, is under constant revision. It improves over time. But that implies that at any particular time it may not be perfect. Again like source code, a project’s documentation may have bugs. That can be terribly frustrating if you are in the midst of a complex installation and come up against an inadvertent miscue in the instructions you are following. You can end up with curious error messages and sometimes no simple way of discerning what has gone wrong or why. So the question on most people’s minds will be: where do you go when the documentation runs out?

[1] See http://open-ils.org/dokuwiki/doku.php?id=system_requirements

> ***Learning point: documentation – even installation documentation – may not always be 100% accurate or fully tailored to your installation; patience may be required.***

Support

Although the documentation for FOSS solutions is always the first place to go for guidance on installation, use, and debugging the software, it is rarely the last port of call. Fortunately FOSS projects make a virtue out of the need for multiple avenues for support. Perhaps the first step in any installation process ought to be a careful revue of the support channels available. Setting a clear plan of which support channels you will use and in which order (thus a graduated plan) might save time in the long run.

> ***Learning point: revue the channels available for support in a FOSS project and how to use them prior to embarking on a complex installation.***

Resources for support vary. Sometimes you will find support from other experts in your team. Or you may rely on support from a single individual as Dibyendra tended to do with Dan Scott. But relying on a single individual for support can create a bottleneck.

Evergreen offers traditional community-based support for its users through public email discussion lists. The *general discussion list* is for all users (technical or otherwise) of the stable released versions of Evergreen. This is the list to use for questions about installation of the current version of the software including raising points about any confusion in the installation instructions found in the documentation. Evergreen has a second email discussion list, the *'dev' list*, specifically for use by those involved in development of the next version of Evergreen. This might also be the best place for feature requests. Typically a message sent to the wrong list, for example a question about the current stable release which is sent to the developers list, will be redirected to the correct list even though it may be the same person who provides the answer on that other list. The reason for this relates back to the documentation effort. Publicly archived discussion lists are a vital source of information about the software. It is very likely that a problem you might experience with the software will have already been encountered (and possibly resolved) by someone else. Therefore, if you search the archive of the email list you may find the solution to your problem even if it has not made it into the official documentation yet.

> *Information on Evergreen's email discussion lists can be found here http://www.open-ils.org/listserv.php The public archive of these lists is easily searchable through a MarkMail interface http://georgialibraries.markmail.org/search/?q=*

In addition to email lists, many FOSS projects utilize Internet Relay Chat (IRC). An IRC channel or chat room provides an opportunity for synchronous interaction. It will typically be used by the distributed team of developers of the software to facilitate rapid exchanges on very technical points. As such, the IRC channel of a FOSS project is not for the faint-hearted. On the other hand, if you are sufficiently technical and have a very specific technical question that you have not been able to resolve either through investigating the documentation, the publicly archived email lists, or through the log files generated by the software, then the IRC channel can be an ideal support mechanism. Indeed it is a good practice to have your IRC client open and to reside in the appropriate chat room when you undertake a significant new software task. It might be just the helping hand you will need in a pinch. Evergreen has its own IRC channel (http://www.open-ils.org/irc.php).

In Dibyendra's case, he was new to the Evergreen software and to the Evergreen community. He was unfamiliar with IRC and unaware of the email discussion lists that the Evergreen community supports. Therefore he restricted his support mechanism to whatever Dan Scott could provide directly via email. Inevitably that led to delays not merely because of the time difference between Nepal and Canada, but also because people have other full-time commitments. In order to maximise the possibility of FOSS support it is necessary to exploit the channels of support set up by the FOSS community in question.

But there is another reason why it is valuable to use the public support mechanisms. Dibyendra may have been running in to problems that others had not yet encountered with the software. If so, then his public identification of these problems would have been a benefit to others who might face the same situation as well as a benefit to developers who might thereby be able to pinpoint a *bug* in the software.

> ***Learning point: use the public support channels as your primary means of support in order to benefit yourself as well as the FOSS project.***

In the end, probably the biggest challenge that Dibyendra faced was his other commitments. A couple of months slipped by in the late summer of 2008 before he was able to dedicate much concerted effort to his Evergreen installation. By September of that year there were already release candidates emerging for Evergreen 1.4. It would have been tempting at that point to simply wait for the final release version of Evergreen 1.4 especially as it was due to have better internationalisation support. But testing, especially of an initial installation, needs to proceed with current stable releases. After all, it is for those releases that the majority of the documentation will be available. Also, FOSS development cycles mean that the software is in a constant state of development. Thus the next stable release will itself be merely a waypoint towards an even better release later. Delaying testing to await the next version does not help in the long run. But even more important is the need for concentrated effort, or rather prioritising effort. A small percentage of one’s time spent on complex software over a period of months may not be as efficient as total immersion for however long it takes to master the software. Of course this is not something always in one’s control, especially when one is faced with multiple technical responsibilities in a small team.

> ***Learning point: piloting requires sufficient dedicated time to enable time to master the key skills required for installing and using the software.***

Eventually of course Dibyendra did complete his installation of Evergreen 1.2.20. With some further assistance from Dan he was ready to begin tackling the data migration issue.

Data Migration

The current MPP catalog consists of a unique MySQL schema designed to support an in-house cataloguing system used to record limited bibliographic data. A significant feature of this in-house system is that it supports versions of the records in Nepalese and English. The challenge for migration is to transform these records into the MARC21 standard used by Evergreen, input them wholesale, and ensure that the full bibliographic information from the records is accessible in the pilot ILS.

Although the database schema used by MPP may be unique, the general problem of moving from a non-standard format to something like MARC21 is not unique. Many libraries face precisely this challenge. Indeed even migration from a proprietary ILS to a FOSS ILS can create a similar challenge. If the set of records in the catalogue is substantial, then migration can become a far more time-consuming task than even the initial installation of a new ILS. Migration of data is one of the main reasons that libraries will seek paid commercial support when migrating to a new ILS.

When commercial support is not an option (or when one is still involved in the testing that might precede a decision to opt for commercial support) you have to use the tools at hand. MarcEdit is the most commonly spoken of tool to aid library data migration (http://oregonstate.edu/~reeset/marcedit/html/index.php). Dibyendra began using MarcEdit in January 2009 in order to begin the process of the migration of MPP's data to the MARC21 format. This was a slow process. It involved learning yet another system.

> ***Learning point: there are so many systems involved in an ILS migration that it may be best to undertake testing and piloting with a small team in order to share the learning and to help with overall support and morale.***

Dibyendra did successfully migrate the MPP data to MARC21 format. In early February, prior to setting the project aside for a time in order to complete his exams, Dibyendra was in the process of importing batches of records into his test installation of Evergreen 1.2.2.0 He was encountering periodic difficulties with the import but was confident that he would be able to overcome this challenge.

Next steps

Clearly this is an ILS pilot that has moved slowly and deliberately. It has a finite and limited set of resources in terms of time and personnel available to it. While it is part of the larger aim of MPP to migrate its catalog of records to a standard format such as MARC21, it would be fair to say that this is not yet a top priority (at least given the current staffing levels at the archive). So Dibyendra, when he returns from his exams, will need to continue his efforts largely on his own. He has a clear plan for how to proceed.

- Support mechanisms:
    - Dibyendra will register on each of the Evergreen discussion mailing lists so that he can post questions to a wider support community.
        - Dibyendra's first questions on the Evergreen lists have already appeared; they were answered promptly.
    - Dibyendra will decide upon an IRC client and seek to develop sufficient IRC skills so that he can, when necessary, call upon support within the Evergreen IRC channel.
        - eIFL-FOSS set up an IRC channel (#eIFL on the freenode network) in order to provide a place to practice using this communications tool.
    - Dibyendra will draft a support path plan.
    - Dibyendra will seek to record his actions (and their effects) in order to aid his learning curve with this software.

- New installation:
    - Dibyendra will move to the latest stable release of Evergreen (1.4.0.4) and the corresponding versions of OpenSRF and the Evergreen staff client. This will allow him to take advantage of the numerous feature benefits in the later version including a better system for importing records.

- Migration and testing:
    - Once the new version of Evergreen is installed Dibyendra will begin importing records and testing.

There is still a long way to go for MPP's testing of Evergreen. Whether or not the final decision will be to migrate to this FOSS ILS, there will have been a great deal of value gained through the testing and piloting process. Most obvious is the skills development for Dibyendra, a key member of the

MPP team. Another benefit is the migration of MPP's catalog data to a standardized format such as MARC21.

Learning Points from the pilot

There have been numerous learning points from this pilot. A few may be unique to MPP or to Nepal, but most have wider application. They will inform our future efforts in promoting FOSS in libraries.

- Piloting requires sufficient dedicated time to enable time to master the key skills required for installing and using the software.
- It may be best to undertake testing and piloting with a small team in order to share the learning and to help with overall support and morale.
- Large-scale systems migration will involve significant demands being placed on the available technical infrastructure
- A wide range of skills, especially Linux system administration skills, will be required.
- Revue the channels available for support in a FOSS project and how to use them prior to embarking on a complex installation.
- Use the public support channels of the FOSS project as your primary means of support in order to benefit yourself as well as the project.
- Be aware that documentation – even installation documentation – may not always be 100% accurate or fully tailored to your installation; patience may be required.
- Migration from one library system to another is not an end in itself; it always serves the larger aims of the library.

For further information on Madan Puraskar Pustakalaya's Evergreen activities, please contact Dibyendra Hyoju dibyendra@mpp.org.np.

For further information on eIFL.net write to info@eifl.net.