

Assessing Evergreen for a Bilingual Academic Library

by

Dan Scott, MIST

Laurentian University

October 2007

Table of Contents

Background.....	3
Assessment criteria.....	3
Bilingual support: Web catalogue.....	3
Evaluation methodology.....	4
Evaluation results.....	4
Recommendations.....	5
Bilingual support: staff client.....	5
Evaluation methodology.....	5
Evaluation results.....	6
Recommendations.....	6
Bilingual support: documentation.....	6
Evaluation methodology.....	6
Evaluation results.....	7
Recommendations.....	7
Conclusion.....	7

Background

Evergreen is an open-source integrated library system that was developed for the Georgia PINES consortium of public libraries. In September 2006, 252 PINES libraries went live with Evergreen. In September 2007, 270 PINES libraries in Georgia were running on Evergreen, with a bibliographic database comprised of more than two million unique recordsⁱ. In October 2007, Evergreen will become an international project as three public libraries in the British Columbia Library consortium are scheduled to migrate to Evergreen as a pilot projectⁱⁱ.

Considerable interest in Evergreen had been expressed at events as the Symposium on the Future of the Integrated Library System at the University of Windsorⁱⁱⁱ, but when SirsiDynix announced that it was discontinuing its popular Horizon library system,^{iv} many more libraries began seriously considering the possibility of migrating to Evergreen or other open-source library systems such as Koha^v. Reflecting the interest in the adoption of Evergreen outside of the its initial monolingual public library environment, this paper attempts to contribute to the discussion by offering an assessment of some of the functionality required by a bilingual academic library.

Assessment criteria

The general criteria used in this assessment of Evergreen are based on the model presented in the Open Business Readiness Rating (BRR) proposed standard for rating open source software.^{vi} A full BRR for Evergreen as a mission-critical component of a bilingual academic library is available in OpenDocument format from <http://coffeecode.net/uploads/reports/BRR-Evergreen.ods> and in Excel format from <http://coffeecode.net/uploads/reports/BRR-Evergreen.xls>. At this time (October 2007), Evergreen was assigned a rating of 3.6/5.0 for the specific context of a bilingual academic library.

This written report focuses on the core functionality required for a bilingual academic library, specifically:

- bilingual support in the Web catalogue
- bilingual support in the staff client
- bilingual documentation

Other functionality that was not considered in depth in this report, but for which initial scores were recorded in the BRR for Evergreen, include:

- academic reserves
- acquisitions
- serials controls

Bilingual support: Web catalogue

One of the original design requirements for the Georgia PINES consortium was to include the ability to display the Web catalogue interface in both English and

Spanish. The internationalization infrastructure, based on the use of XML entities, was in place early in the project. *XML entities* are placeholders for content that can be mapped to one or more translated versions of that content. However, to meet the implementation deadline of September 2006, the development team was forced to finish the 1.0.0 release with many strings hard-coded into the software.

Evaluation methodology

As the bilingual support infrastructure was already in place, the primary evaluation of the bilingual support of the Web catalogue consisted of replacing the remaining hard-coded strings with XML entities, translating the XML entity definition file, and determining where the Web catalogue depends on strings or monolingual functionality outside of the easily translated XML entity definition file. In addition, the ability to support different collating sequences (language-specific sort orders) and the way in which search queries handle characters outside of the Latin-1 character set was considered.

Evaluation results

As part of this evaluation, 100% of the static content for the default Web catalogue interface has been converted to entities. Approximately one week of development time was required to complete this work. In addition, a professional translator created a complete set of French entities so that the evaluation could be completed. All of this work has been contributed back to the Evergreen project. The current method of switching languages is URL-based, so <http://dev.gapines.org/opac/en-US/skin/default/xml/index.xml> presents an English interface based on the **en-US** portion of the URL, while <http://dev.gapines.org/opac/fr-CA/skin/default/xml/index.xml> presents a French (Canadian) interface based on the **fr-CA** portion of the URL. To switch languages, the user currently must either manually replace the locale in the URL with the desired locale, or follow a static link to start a new search session in the new language.

The collating sequence for Evergreen is currently determined by the locale used to initialize the PostgreSQL database. This limits the available collating sequences for a given Evergreen installation to a single locale; however, the only significant difference between French and English collating sequences is that the French collating sequence uses the right-most accented character in a word to determine sort order.^{vii}

Accented characters in search queries are currently ignored by Evergreen, so é, è, and e are all treated as the same character. This behaviour can be changed, but it requires a change to both the indexer and to the search query parser.^{viii} I estimate the work required to complete this customization at less than a day of development time.

There is a small set of dynamic content, including item types and library names, drawn directly from the Evergreen database. As the database currently uses a monolingual schema, these aspects of the Web catalogue are not fully bilingual. A

multilingual database schema is planned for the Evergreen 1.4 release.

Evergreen invokes spell checking support based on the freely available GNU aspell^{ix} spell checker and its accompanying standard English dictionary when less than ten hits are returned for a given search. I estimate the work required to customize Evergreen to offer spelling suggestions based solely on a French dictionary at less than an hour of development time. However, designing spell checking functionality that supports the user with meaningful suggestions based on both French and English dictionaries would be significantly more difficult. In addition, tests of the current spelling suggestions show that the system often suggests terms that lead to no hits. A different approach based on fuzzy term matching and frequency analysis of the terms in the corpus of bibliographic records in the system may be a more fruitful avenue for both monolingual and multilingual systems.

Recommendations

- Implement a more sophisticated means of switching locales, such as detecting the preferred browser language setting, enabling a language preference in user account, and adding a language selector to every page that maintains the user's context when they switch languages.
- Implement the multilingual database schema to support dynamic content in multiple languages.
- Investigate the feasibility of implementing spelling suggestions based on fuzzy term matching against the bibliographic records in the system.

Bilingual support: staff client

The Evergreen staff client is built on the Mozilla XML User Interface Language (XUL)^x platform – the same infrastructure used to make the Firefox Web browser available on Windows, Linux, and Mac OSX operating systems in dozens of different languages. XUL includes multilingual support in the form of XML entities for XML files and, for JavaScript files installed on the same workstation as the XUL application, in the form of string bundles. The Evergreen staff client uses a small core of locally installed XUL and JavaScript files to provide offline support, and relies primarily on XUL and JavaScript files served from a remote server for the bulk of the functionality. This approach enables upgrades to the Evergreen staff client to be performed without requiring every workstation to install an updated version of the staff client.

Unlike the catalogue, the requirements for the Evergreen staff client did not include the ability to support multiple languages. The resulting implementation therefore contained mostly hard coded English strings.

Evaluation methodology

A small but representative portion of the Evergreen staff client was converted to entities and XUL string bundles to determine the feasibility and estimated cost in

development time of extending that support throughout the entire client.

Evaluation results

Most remote XUL files currently use a hard coded English locale (en-US) in the DTD definition. This needs to be converted to an Apache server side include variable (`${locale}`) so that the locale can be changed dynamically by the server in response to a changed user preference. This is a relatively trivial, easily automated change.

The initial approach of using standard XUL string bundles for multilingual support in JavaScript quickly turned out to be unusable, as the XUL architecture prevents the use of remote string bundles. After discussing this problem with the core Evergreen developers in late July, 2007, they developed a custom drop-in solution (named "message catalogs") for the problem and deployed the solution in August, 2007.

Approximately 15% of the staff client code was converted to multilingual support. This was enough to determine that the new approach was workable, although some edge cases (such as encoding escaped characters like carriage returns and tabs) still need to be corrected. I estimate that four weeks of development time would be required to complete the conversion to a multilingual client.

Recommendations

- Release the message catalog internationalization framework as a separate XUL project. This will expose the code to more use cases and attract contributions of additional functionality and bug fixes.
- Complete the conversion of the staff client code to multilingual support.

Bilingual support: documentation

Evergreen suffers from a lack of formal documentation, a common problem with open source products. The Evergreen wiki at <http://open-ils.org/dokuwiki/doku.php> is currently the main source of documentation. The wiki contains a significant amount of technical documentation, training documents that were written and contributed by PINES members, and a small but steadily growing collection of user-contributed documentation.

Evaluation methodology

The documentation was evaluated first using the BRR criteria of determining whether an installation guide, deployment guide, and user guide were available, and whether product users were able to contribute to the documentation. Then the availability of the documentation in French, and the ability to translate documentation, was considered.

Evaluation results

The wiki currently contains a significant amount of installation information, including step-by-step guides for two different Linux distributions. There is a lack of documentation for administrators of Evergreen systems, however, in deployment subjects such as migrating bibliographic and patron data from another library system. The PINES training documentation covers some of the content required for a user guide, but there is no online help for the staff client or catalogue.

None of the Evergreen documentation is currently available in French. The wiki lacks a good mechanism for supporting translations of the documentation.

In an attempt to address both the lack of formal documentation in the form of deployment guides, user guides, and online help, and to provide a framework that would support synchronizing translations of the documentation into French and other languages, a table of contents written using the open standard for technical documentation, DocBook^{xi}, was added to the Evergreen code repository in September 2007 and documentation for generating XHTML and PDF output from the DocBook source was added to the Evergreen wiki. A call for documentation writers was issued to the Evergreen mailing lists, with a number of positive responses from the community indicating that they would be willing to contribute to the official documentation.

Recommendations

- Invest several weeks of effort in developing a seed set of documentation that exemplifies the approach, voice, and formatting that is desired for the complete set of documentation.
- Make the ground rules for contributing to the documentation clear and simple.
- Set up a publicly visible nightly build of the DocBook-based documentation in English to provide rapid feedback on contributions and encourage further contributions from the community.
- Investigate the possibility of reusing the same DocBook source to generate online help in XUL format.
- Extend the documentation build to provide a per-language translation framework.

Conclusion

While the Evergreen open source library system is not ready for immediate adoption by a bilingual academic library, a one-time effort of approximately 5 weeks would be required to provide a bilingual Web catalogue and staff client with circulation and cataloguing support in the 1.4.0 release time frame.

However, significantly more effort will be required to create and maintain a comprehensive set of documentation in both languages. This report underscores

the importance of building a strong community willing to contribute documentation and an infrastructure capable of supporting that community's efforts.

Finally, while the effort required to implement academic reserves, acquisitions, and serials support has not been considered as part of this report, the BRR for Evergreen strongly suggests that additional attention needs to be paid to these areas of functionality.

- i Lajeunesse, Brad (2007). *Evergreen's first birthday*. Retrieved October 9, 2007 from <http://open-ils.org/blog/?p=99>
- ii BC Library (2007). *Project management framework: update: July 6, 2007*. Retrieved October 9, 2007 from <http://pines.bclibrary.ca/resources/project-management-framework/project-management-framework>
- iii (2006). *The Future of the Integrated Library System* (programme). Retrieved October 9, 2007 from <http://infoservices.uwindsor.ca/ils/wp-content/uploads/2006/11/ils-symposium-final-pgm.pdf>
- iv Hane, Paula J. (2007). *SirsiDynix to Consolidate Products on New Platform*. Retrieved October 9, 2007 from <http://newsbreaks.infotoday.com/nbReader.asp?ArticleId=35701>
- v Breeding, M. (2007). *The sun sets on Horizon*. Retrieved October 9, 2007 from <http://www.librarytechnology.org/blog.pl?ThreadID=23>
- vi SpikeSource, Center for Open Source Investigation at Carnegie Mellon West, and Intel Corporation (2005). *A framework for evaluating open source software*. Retrieved July 9, 2007 from <http://www.openbrr.org/>
- vii Davis, M., and Whistler, K. (2006). *Unicode Technical Standard #10: Unicode Collation Algorithm*. Retrieved August 22, 2007 from http://unicode.org/reports/tr10/#French_Accents
- viii Rylander, M. (2005, May 5). *Re: find more like this one*. Message posted to code4lib electronic mailing list, archived at <http://www.infomotions.com/serials/code4lib/archive/2005/200505/0158.html>
- ix GNU Aspell. <http://aspell.net/>
- x XML User Interface Language (XUL) Project. <http://www.mozilla.org/projects/xul/>
- xi The DocBook Project. <http://docbook.sourceforge.net/>