Kuali OLE at the University of Chicago Library

By Frances McNamara, Stuart Miller, and Tod Olson

LYRASIS

Kuali OLE at the University of Chicago Library

by Frances McNamara, Stuart Miller, and Tod Olson

LYRASIS

First edition: September 2014
This case study was funded by a grant from the AndrewW.MellonFoundation and is published under the terms of the LYRASIS/Mellon Intellectual Property agreement.
Cover graphic created by Tagul.com.

PDF version generated by LibreOffice 4.3.2.2
ePub version generated by writer2epub 1.1.28 and Calibre 2.3.0

This document is also published on the FOSS4Lib.org website.

Table of Contents

Abstract

Purpose

This case study documents the processes at the University of Chicago Library (Library) that ultimately resulted in the decision to replace the Library's proprietary systems – SirsiDynix's Horizon, Innovative Interfaces' Millennium Acquisitions and Serials Solution's AquaBrowser – with the open source VuFind user interface and Kuali OLE (Open Library Environment).

Design/Methodology/Approach

This study is based on internal documentation and interviews with Library staff, complemented by links to documents available online.

Findings

This study will seek to highlight the successes, failures and lessons learned during the replacement project.

Research Limitation

The Library is not moving its new systems into production until July 2014. Final determination of success depends on the Library's planned goal of shutting down its legacy systems by January 2015.

Practical Implications

Other academic libraries considering open source systems may find the University of Chicago Library's experience of interest.

Originality/Value

The Library will be one of the two first adopters of Kuali OLE. This study may help other institutions evaluate and plan for similar system migrations.

Background and Origins

The University of Chicago Library (the Library) houses a large research collection of 11.9 million volumes in six libraries on the university's main campus in Hyde Park, a neighborhood of Chicago. Details on the size and content can be found at http://www.lib.uchicago.edu/e/about/.

Basic facts about the University of Chicago (UChicago) can be found at http://www.uchicago.edu/about/. The Library serves a student body of 5,692 undergraduate and 9,502 graduate, professional, and other students; 2,186 full time faculty; and 9,500 staff members at the University of Chicago Medicine (details can be found at http://www.uchospitals.edu/about/fact/hospitals-sheet.html).

In the early days of library automation beginning in the 1970s, the Library developed its own, mainframe-based system (Library Data Management System or LDMS), run on an IBM machine housed in the campus computing data center. When it was decided to retire the mainframe, the Library concluded that purchasing a commercial system was the most viable option and migrated to the Horizon system in 1995. (Horizon was originally owned by a division of Ameritech, which sold the software company to Epixtech, eventually changing its name to Dynix, bought by Sirsi to become SirsiDynix, the current owner.) The acquisitions portion of Horizon never completely met the needs of the Library, so those functions were migrated in 1997 to what is now Innovative Interfaces' Millennium system (although the Library uses only its acquisitions functions). Providing users with information regarding on-order titles was accomplished by regularly scheduled exports from Millennium that were then batch imported into Horizon.

Like any client-server system, Horizon used a relational database management system to handle data storage functions; in this case, Sybase was the product and its cost was typically bundled into the overall cost of Horizon. However, the Library was able to obtain better pricing through the University's existing campus-wide Sybase license. The University's IT Services unit also negotiated the annual Sybase maintenance cost for the Library. This allowed the Library direct access to Sybase support, which was often helpful for debugging and local development efforts.

As with the LDMS hardware, the Library entered into an agreement with IT Services to run Horizon within the university's enterprise environment to take advantage of its Storage Area Network, Tivoli Management System for tape backups, and system administration support. Although the Library still had to pay for its hardware and for the support services, it was far more economical than hosting hardware within the Library, which would have resulted in the need to add at least one full-time system administrator. The Library meets regularly with IT Services to deal with any issues. The Library has direct, root access to its Horizon servers and installed software upgrades in coordination with SirsiDynix and IT Services.

The Library also maintained separate test and reporting environments of the Horizon software that allowed for installation and testing of new software releases prior to moving them into production. This could be done under both the Horizon and Sybase software licenses at no additional charge. This also allowed Library staff to develop specialized, local applications "around the edges" of Horizon, frequently without the vendor. For this reason, Horizon, although proprietary, was a comparatively "open" system as it was installed at the University of Chicago.

In contrast, Innovative Interfaces' Millennium was a black box. It resided on its own server in the Library's system environment with its own tape backup. Upgrades and customizations all had to be performed by vendor support staff, who required direct access to the system. While the Library would have preferred to have a test environment (as it did with Horizon and Sybase), Innovative Interfaces charged a comparatively high price for a second copy so it was decided not to try to support a test environment.

The Library had always been aware of the cost of supporting multiple systems and it had always been clear that continued support for multiple systems was not in the Library's best interests, either functionally or financially. So when SirsiDynix began work on a completely new system called Horizon 8.0 (also known as "Corinthian") to replace the existing Horizon AND include desired acquisitions features, the Library agreed to be a development partner, working on specifications and testing with vendor staff from 2005 to 2007. Library staff were trained on Corinthian in the summer of 2006, but the lack of some critical features caused a postponement of the migration. During this period, ownership of SirsiDynix changed twice, but there were repeated commitments to finish Corinthian. However, in February 2007, SirsiDynix announced it would stop Horizon 8.0/Corinthian development. Experience gained by library staff working on this project was later helpful in work on the Kuali OLE project. During the Corinthian project, the Library contributed to and reviewed functional specifications. They also participated in testing of the software during development. They became familiar with the bug tracking JIRA system used to communicate with software developers to fix problems, and they became used to the process of frequent upgrades to a test system to test fixes. Weekly calls with the product manager and various project management tracking tools were used. Also, analysis for data conversion to that system was done and data clean up issues that needed to be addressed prior to migration to any other system were identified.

Horizon had several online public access catalog (OPAC) modules over the years until 2002, when the Horizon Information Portal (HIP) was introduced to replace all earlier modules. Horizon Information Portal (HIP) became the sole end-user interface for the Library. By 2006, there were technical advances in other library search retrieval systems. The Library began an investigation of newer, "faceted browse" search systems. In November 2006, the Library issued an RFP and eventually selected AquaBrowser after an extensive review (see http://www.lib.uchicago.edu/staffweb/depts/ils/projects/faceted-browsing/ for more details).

While providing many advances over HIP, Lens (the locally adopted name for AquaBrowser) was never a full replacement due to its lack of functionality, e.g., no support for non-Roman searching, so the Library continued to support both HIP and Lens. Then, AquaBrowser was bought by Serials Solutions, which announced plans to offer the product only as a hosted service without any of the many customizations that the Library had already made to it. There were no plans to support non-Roman searching on the locally hosted version of the product. Also during this period, the Library made the EBSCO Discovery Service (EDS) available to users on a trial basis; their favorable reaction led the Library to select it as a replacement for the Ex Libris Metalib system which was a Z39.50 based system that sent searches across multiple databases and federated the results into a single list.

By the time the Library began to consider system replacements, the public catalog functions had been to some extent abstracted from the ILS to a discovery layer. Even HIP was a separate system on a separate server to which records were passed from the Horizon system to be indexed and displayed. AquaBrowser also worked by indexing records exported from Horizon and using servlets to reach into Horizon for some dynamic information such as item status.

However, supporting two separate user interfaces, HIP and Lens, plus Horizon and Millennium, was becoming increasingly onerous. In addition, with the demise of Horizon 8.0/Corinthian, the Library decided not to move beyond the Horizon 7.5 release; any new development planned for HIP would require moving to new Horizon releases. This effectively meant that the Library would not be able to install any new versions of HIP. Changes in the ownership of AquaBrowser and terms of service had made its continuation unattractive as well.

Integrated Library System Renewal

In addition to the general undesirability of supporting multiple systems, the technical platform for Horizon became more expensive and more difficult to support as time went on. The Sybase license required an AIX platform and while that had previously been used extensively by many other university applications, thereby reducing everyone's annual maintenance, most of those applications migrated to more cost-effective hardware and software. As a result, the Library's annual maintenance for Sybase doubled. The only other relational database management system supported by Horizon is SQL Server and this was not considered a good choice for the size of the Library's database or for the environment where we wanted to run it.

Another problem was that the Horizon database was never upgraded to Unicode and this caused an increasing number of issues as the Library has an extensive collection of titles in non-Roman alphabets. The Library had to run imported record files through conversion programs; while these generally worked, the pre-Unicode standards applicable to Horizon did not always result in completely correct translations. In addition, SirsiDynix, following the demise of Horizon 8.0/Corinthian (which was Unicode based) announced that it would NOT convert Horizon 7.x to Unicode. The only option was to move to the vendor's Symphony system, which was Unicode-compliant. However, that system was – following an assessment by Library staff – more or less functionally equivalent to Horizon 7.x; the complexity and cost of a system migration could not be justified if the new system was merely "equivalent" to the old, so this was not considered a good solution.

Environmental scan of commercial systems

Following the halt of Horizon 8.0/Corinthian development, the Library began an immediate scan of the ILS environment in early 2007. To do a quick survey of possible replacements, the Library's systems staff arranged for conference calls with the University of North Carolina at Chapel Hill to discuss their implementation of Innovative Interfaces' Millennium and with Boston College Library for a discussion of its implementation of Ex Libris' Aleph system. A tabulation of the results of those discussions is at: http://www.lib.uchicago.edu/staffweb/depts/ils/projects/ilsreplacement/systemsevaluation.html.

Contacts were also made with Oxford University and New York University about their work with VTLS on its new system, but the discussions led to the conclusion that this was not a production-ready option.

Of all the options available at that time, only Aleph seemed reasonable given our requirements, and Ex Libris did an on-site, day-long demo for the Library. Its preliminary, informal price quote was surprisingly low, but a later formal quote was significantly higher. In the end, the Library could not justify such a significant investment on a system that, while meeting most of the functional requirements, would not give the Library a more technologically up-to-date platform than it already had.

Ex Libris invited the Library to be a development partner on what eventually became Alma, its next generation ILS. However, the Library was reluctant to commit to another vendor's new system development project. The fact that Ex Libris' ownership changed in 2007 (bought by a private equity firm, as SirsiDynix had been) was also not encouraging. So the Library declined to participate.

By this time, other developments were suggesting other alternatives.

Requirements documents from Library functional groups

Around 2007, open source ILS products were just beginning to attract attention as alternatives to the commercial products. Evergreen and Koha were the open source library systems that seemed to have the most promise, and information about both were actively collected by the Library. Since systems staff were aware that any new ILS would undoubtedly have certain functional gaps, groups of Library staff were asked to identify the core functional requirements for their area. Work had already begun on this analysis during the Horizon 8.0/Corinthian development as it was necessary to enumerate critical requirements in order to verify the system would be able to replace existing systems. Before doing a "gap analysis" on the existing open source systems, we asked staff to formalize these lists of requirements. These lists can be found in a chart that links to Word documents at: http://www.lib.uchicago.edu/staffweb/depts/ils/projects/ilsreplacement/.

Technical investigation of open source

Investigation of Evergreen proceeded with discussions with its developers and some academic libraries including McMaster University and Project Conifer, a group of academic libraries in Ontario that were considering developing Evergreen sufficiently to enable its use in academic libraries. While Evergreen was developed by and for a consortium of Georgia public libraries, it seemed to have the potential of being scalable and workable as a platform for development of additional features. It was promising enough and mature enough that the Library's systems staff installed the software and loaded a full copy of the Library database to allow staff to do a gap analysis. Results of that analysis can be found in the chart at: http://www.lib.uchicago.edu/staffweb/depts/ils/projects/ilsreplacement/ with reports by various functional groups. *Note that there are comments that sometimes give an assessment of what it would take to remedy a gap. [A gap assessment of Koha was not done to this level of detail so that column does not contain this type of report].* Evergreen looked promising but it definitely lacked some features essential for academic libraries. For instance, academics have a changing patron file that needs to be updated to reflect enrollment and library privileges; fixed due dates are common; and the capability to recall items is often required. Interface with various university systems is also needed, e.g., the ability to export voucher data to university payment systems.

The library sent Stuart Miller to the VALE (NJ) "Next Generation Academic Library System Symposium" in March 2008, which discussed Evergreen, Koha and some other library open source systems, including VuFind. It was openly acknowledged that SirsiDynix's decision to abandon Horizon 8.0/Corinthian had created a sudden increase of interest in open source products among libraries.

Contact was also made with WALDO, a consortium of fifteen academic libraries in the New York area (the largest being St. John's in Queens, NY) that was implementing Koha. LibLime, the support company and the North American release manager for Koha, had completed several enhancement projects funded by WALDO; St. John's was then in production. A conference call with Joshua Ferrero and John Rose of LibLime was held on April 10, 2008. (See "Some Basics on Koha Discussion with LibLime April 2008" on http://www.lib.uchicago.edu/staffweb/depts/ils/projects/ilsreplacement/.)

Koha's drawbacks were found to be primarily technical – there were questions about its scalability to much larger databases. No large ARL library had been involved, and there was a suggestion the software might need to branch to support ARL library workflows. Library systems staff continued to monitor developments, but it was eventually dropped as an option. It was considered better to wait for changes to support academic library work to be completed before expending the effort for a complete evaluation that involved installing a local copy with the full library database. By the time those changes were available, the library had already decided to participate in the Kuali OLE project.

In 2009, the Library also thought it should review the state of the new OCLC Worldcat library management system; a summary report can be found at: http://www.lib.uchicago.edu/staffweb/depts/ils/projects/ilsreplacement/. Due to many non-existent features at that time, a detailed gap analysis was not thought to be necessary. In addition, there were doubts that this new product could achieve acceptable performance levels, and the Library's past experience with the quality of OCLC support and services led to skepticism about its ability to provide even an acceptable level of support for a mission-critical service such as an ILS. The fact that the library is open until 1 AM weekdays and is heavily used late at night and on weekends tends to influence decisions about whether trying to use a hosted service makes sense. Library systems staff do provide support nights and weekends, and it was felt this level of support would be difficult to afford via a hosted service.

Invitation to participate in OLE project phase 1

Meanwhile in 2008, the Andrew W. Mellon Foundation funded a project to design an "Open Library Environment" and work began in August 2008. University of Chicago participated in this project, along with several other large academic libraries, and hosted one of the regional meetings in December 2008. Documents from this meeting including a useful OLE Overview presented by Jim Mouw, Associate Director for Collection Services, University of Chicago Library, can be found at: http://www.lib.uchicago.edu/staffweb/depts/ils/projects/ilsreplacement/OLEUChicworkshop.html. Work on the first phase of this project included training in Business Process Management and production of Tasks and Process Maps for each module of a library system. This resulted in a design document that was presented to Mellon in early summer of 2009.

Interest in OLE also came from Greg Jackson, then director of the University's IT Services. The OLE Project had discussed the need for a new ILS to work at an enterprise level, e.g., a library's need for patron data would be satisfied by being able to pull data from a campus database rather than having to maintain a set of patron records like all current ILS systems. Inspired by a conversation with Library Director Judi Nadler, Jackson raised this issue at a meeting of the Common Solutions Group (CSG – http://www.stonesoup.org/ – a network of campus CIOs). Clearly, an "enterprise level" ILS capable of direct interface with other campus applications would require cooperation and input beyond libraries. CSG, while interested, did not "adopt" this as something that the group could fully support, not because it was undesirable, but because most institutions were just not ready for it. Any OLE design would need to recognize this. This did prompt OLE to think that repurposing what already existed might also be the most practical way to approach a new ILS, e.g., taking Evergreen and adding development suitable for academics might be a possibility (although after review, OLE decided to use Kuali Rice instead). Also, OLE began to consider whether or not it should ultimately become a Kuali Foundation project, partly in order to facilitate cooperation with other open source campus application developers.

Decision to become Kuali OLE partner

At that point, a "build" project for OLE had not been funded, but was in the offering. The Library was concerned at that time that the Horizon system might not be supported by SirsiDynix beyond 2013. *[Note: As it turned out, Horizon is still supported by SirsiDynix, and there is, as of March 2014, no announced "end of life" for Horizon.]*

Because the library was not ready to choose a true replacement system for the ILS, a "bridge" solution was proposed in October 2009. The Bridge to the Future recommendation (http://www.lib.uchicago.edu/staffweb/depts/ils/projects/ilsreplacement/) summarized the possible options: to implement Evergreen; stay on Horizon as is; or upgrade to Sybase 15 and upgrade to Horizon 7.5.1. It was later found that the cost for the Sybase license was half what was originally estimated, and that final option was eventually implemented. An added concern for the Library at that time was the construction of the Mansueto Library planned to open in 2011. The Mansueto Library was built to house lesser used portions of the collection in a structure that is physically connected to the Regenstein Library. Transfer of parts of the collection to this library allowed newer materials to continue to be shelved in the open stacks, while avoiding the problems associated with offsite storage. There was a requirement that the automated storage and retrieval system in Mansueto interface with the Library system to allow users to seamlessly request items in storage. This would have to be done first for Horizon and then with whatever its replacement would be.

In order to apply for Mellon Foundation funding to build the OLE software, it was necessary for a group of libraries to agree to be founding partners and to contribute matching funds. In 2009, the Library decided to become a founding partner, and the grant proposal to build the OLE software was funded by Mellon to run from July 2010 to June 2012 (additional funding was later secured from Mellon through 2014).

The OLE project joined the Kuali Foundation in December 2009. The Kuali Foundation provides the legal and financial framework that is needed to sustain an open source project, and because all of the Kuali projects that come out of the academic environment, it is set up in a way that works well for an academic library.

As with all previous efforts concerning system replacements, the Library's systems staff recruited assistance from the University's IT Services department in considering open source software. The Kuali project was already being monitored because of the university systems that fall under its umbrella. While the University of Chicago had no plans to implement other Kuali software at this time, they are considered viable options. Some work was moving forward by the identity management staff in IT Services to discuss a potential open source identity management system that might be developed.

Decision to become an early adopter

With the decision to become a full partner in the "build" of OLE, the Library had a plan to implement OLE in the summer of 2013. It was thought that the first usable version of OLE would be released in the summer of 2012, and then it would take roughly a year to install, customize, convert data and integrate the system. *(Note that a full year was projected because the software would be so new; later implementers should be able to do this in less time.)* Not too surprisingly, there were development delays and setbacks, so the Library's projected implementation date is now July 2014.

The Library was willing to become one of the first OLE adopters for a number of reasons. First, the clock was running out for support for the hardware and software of the legacy systems. In addition to annual support costs for the two systems and the increasing costs for the Sybase license, the Horizon AIX server would soon need to be replaced. To migrate to a more supportable hardware platform would require another license upgrade to run on Linux and that would be costly. As it turned out, the Library had to replace the HIP server in 2013 because of its advanced age, a project the Library hoped to have avoided.

Second, the Library, as an early adopter, would be able to work directly with the developers to ensure that OLE would work with its large database and that its essential requirements (referred to as its "drop dead" list) would be met. In addition, the Library was extremely eager to move to a Unicode-based system; this basic lack in Horizon was contributing more and more to the overhead support costs.

While any system migration is challenging and complex, the Library believes a migration here is somewhat easier due to a variety of factors. For one, Chicago is far more centralized than many comparable university libraries; all six libraries on campus have all used the same system for many years, and all libraries report into one management structure with one director. In addition, there is a single identity management system for users. Library staff are accustomed to participating in review of specifications and beta testing, having done so for Horizon in 1995 and more recently for Horizon 8.0/Corinthian and AquaBrowser. Staff had long been involved in planning for system migration and many were acting as subject matter experts writing functional requirements for OLE.

Another factor simplifying an OLE implementation was that the Library's legacy systems had never included Electronic Resource Management (ERM). While OLE includes ERM, it is possible to migrate to OLE and implement the ERM features later.

Discovery Layer Online Catalog

From its inception, Kuali OLE decided NOT to include an online catalog module in its design, due to the wide availability of several open source interfaces such as Blacklight and VuFind, two of the most popular. It was assumed that Kuali OLE would support any necessary protocols to allow for connectivity and the ability for users to access "my account"-type features (e.g., items checked out, renewals, lists of requested items, updating addresses, etc.).

So beginning in 2011, the Library began a technical evaluation of possible user interfaces as well as initiating an assessment of user requirements.

Under the leadership of Tod Olson, Systems Librarian, Library systems staff did an investigation of Solr based on the open source systems Blacklight and VuFind to assess the technical feasibility of implementing one of these systems with a Kuali OLE-type database. In January 2012, this produced the Solr Catalog Technical Report (see http://www.lib.uchicago.edu/staffweb/groups/disctools/index.html). To quote the summary findings:

> · Implementation language (user interface level). Blacklight is a Ruby on Rails application. We have no local experience with this platform, and there is a significant learning curve. VuFind is a PHP application, the Library has in-house experience with PHP and there is a lot of support on campus for PHP.
>
> · VuFind has more of our critical features out of the box, notably browse indexes (title, author, subject, and call no.), a built-in architecture for integrating live circulation data, and a built-in authentication framework. Blacklight is working to close this gap, but currently Blacklight supplies fewer needed features than VuFind.
>
> While either platform would be suitable for building our public front-end for Kuali OLE, VuFind appears to be the better match for the Library. VuFind is the recommended platform.

Note that the recommendation was not so much to implement VuFind as is, but to use it as a platform to develop a user interface for Kuali OLE and replace both HIP and Lens.

Meanwhile, a group of library staff was appointed to collect information from users about their needs and desires for an ideal user interface. Following the agile development concept of collecting user stories, the Library did just that with a series of activities involving a cross-section of Library users. Findings were published in a report (see http://www.lib.uchicago.edu/staffweb/groups/cus/). A quote from the full report describes the methodology:

> …With the User Stories method, investigators seek to solicit statements of user need. These stories are typically written in plain, rather than technical, language. Stories are constructed so as to avoid complex or interdependent requirements. Stories do not specify a particular solution, but rather they seek simply to describe the need.
>
> The project team began by reviewing existing sources of data to identify user stories. Sources reviewed include comments from multiple LibQual and Library surveys, user email comments in Knowledge Tracker and Bugzilla, requirements documents from Lens development and from a requirements list developed by Stanford, and from prior usability studies. Approximately one hundred unique stories were drawn from these sources. These stories were categorized, and they informed the design of interview questions and research instruments that were used in subsequent data collection.

Beyond the mining of existing data sources, the project team used several methods to collect data specifically for this study. Library staff conducted twenty individual and group interview sessions, involving a total of twenty-seven participants. Seven interviews were conducted by bibliographers, who recruited faculty from contacts. The remaining interviews were conducted with students from a variety of disciplines and programs, and with one College graduate working as a clinical researcher. These participants were recruited using ads on the UChicago Marketplace site, and on the Library web site, and the Library offered a $15 Amazon.com gift card as an incentive.

Taking these two reports together led the Library to select VuFind as its new "front end" with Kuali OLE as the back end to replace both HIP and Lens. It was decided to introduce VuFind as a "beta test" to library users in early 2014 using the Horizon database. User feedback would be used to fine tune the product. In the meantime, VuFind would be tested against Kuali OLE to ensure that once the latter went into production, VuFind would work with it. Since they would already be familiar with VuFind, library users would not even notice the switch on the "back end" and Library systems staff would have some breathing room between implementing Kuali OLE and implementing VuFind.

Chicago Migration

The Library is currently in the midst of our migration project with the goal of bringing Kuali OLE and VuFind into production in July 2014. VuFind was introduced to the public in January 2014 as a "beta" system that was connected to the production Horizon system. It was decided to wait for the OLE implementation to make VuFind the production public interface, since that implementation was pending and it is usually preferable to implement substantial changes over the summer in an academic environment. While VuFind was essentially production ready against Horizon in spring 2014, actual migration to it as the production OPAC and retirement of HIP and Lens was delayed until summer 2014 to coincide with the OLE migration.

In addition to the internal processes to migrate systems, there is a substantial amount of staff time contributed to the open source Kuali OLE project itself. In planning for staff resource allocation, it was necessary to recognize commitment to the OLE project as well as to internal activities. Appendix A lists the required staff contributions by partner sites during this phase of the project. Participation in the project required Library staff to become proficient in use of the project collaboration tools, including WebEx, Google Docs and JIRA. Some staff were trained in creation of Selenium scripts although that effort was abandoned eventually due to the difficulty of the constant changes to database structures and screen displays during the very active development cycle for version 1.5. The scripts could not be maintained until more of the development was complete and the User Interface was changing less frequently. During this part of the development the OLE central project, QA staff were charged with doing the scripting where feasible.

Functional Migration Activities

To facilitate the Library's migration process, the University of Chicago Integrated Library System unit (ILS) established an ILS Migration Steering Committee (IMSC) made up of key decision-makers in the Library. The IMSC reports directly to the Library's Administrative Committee (AdCom); two members of the IMSC are also members of AdCom. The IMSC's charge and duties are as follows:

> IMSC, under the guidance and direction of ILS, exists to make decisions about any matters pertaining to the migration from Horizon and Millennium to Kuali OLE. These matters include, but are not limited to:
>
> * pre-migration data cleanup
> * data mapping from old to new systems
> * data archiving of historical information NOT to be migrated to OLE
> * OLE configuration settings
> * testing preliminary and final release versions of OLE
> * developing training programs for staff
> * training staff

* scheduling
* internal communications

IMSC is the decision-making body for the migration process as a whole and will meet as needed to discuss cross-functional concerns, hear status reports from lead members for the various functional areas, and/or resolve specific functional issues referred to it. It, in conjunction with ILS, will identify and assign migration-related tasks and projects. The IMSC will document its activities on Staffweb and/or Basecamp and provide regular status reports to AdCom (schedule TBD). AdCom provides IMSC with input on any general matters referred to it, and may raise questions, intervene on any issue, or give directives as needed.

IMSC members acting as leads for a functional area are expected to speak for all staff stakeholders in that function (regardless of work unit). As such, each lead member will form an informal subgroup of stakeholders for the purpose of soliciting input and reaching decisions on migration matters related to the functional area. Should a subgroup be unable to reach decisions on any matter, IMSC as a whole will make the decision. Subgroups will meet on their own (with the ILS liaison) as needed and may be invited to meet with the full IMSC as required.

IMSC will also coordinate its work as needed with the Discovery Tools Group as it works to implement the Library's new user interface in conjunction with OLE. The Web Program Director as an IMSC member-at-large will be the Group's liaison.

Following OLE implementation, it is envisioned that IMSC will be replaced by a group to advise ILS on implementation of new OLE releases, define and prioritize enhancements, etc.

Duties

All members will be expected to:

* attend meetings (frequency to depend on nature/scope of pending/current projects)
* respond to questions, make recommendations, etc., as requested by ILS concerning specific matters related to the migration
* assist ILS in communicating major decisions to staff
* perform and/or coordinate assigned tasks

In addition, lead members for a functional area will be expected to:

* represent the requirements/concerns of staff stakeholders of the functional area (regardless of library work unit)
* recruit staff stakeholders
* lead discussions with stakeholders on migration matters and make decisions as required
* perform and/or coordinate assigned tasks such as data cleanup, review of data mapping documents, test software, etc.
* determine training requirements for the functional area and assist ILS in identifying trainers and developing training classes

"Lead members for a functional area" refer to those IMSC members who lead "working groups" in the following areas:

1. Cataloging
2. Acquisitions
3. Circulation
4. Serials Receiving
5. Reporting

The working groups' members are key staff members in their respective functional areas and work with a designated ILS staff member on any matters related to moving data and operations from Horizon and Millennium to Kuali OLE for the respective functional areas. VuFind implementation is managed by a separate group that includes technical support from ILS; insofar as a migration of VuFind from Horizon to Kuali OLE is concerned, ILS' role is primarily technical, although the circulation working group will be involved in testing "my account" functions when these are moved to VuFind/Kuali OLE.

Most of the actual work of performing gap analyses, testing, developing training, etc. is done by members of the IMSC working groups. It is important to note that since ALL Library units use one system, we have explicitly made clear that the working groups "represent the requirements/concerns of staff stakeholders of the functional area (regardless of library work unit)". This is an effort to make clear that while individual library departments performing the same function (e.g., the Library has four cataloging departments) may have differing policies or procedures, use of the ILS must be the same.

One of the most important tasks of the IMSC is to ensure that staff receive the necessary training in Kuali OLE that includes both the instruction on how to use the new ILS AND any changes to policies/procedures/workflows that the new system will require. The IMSC has adopted a training plan included as Appendix B.

Details on our migration – along with links to the IMSC charge, training plan and other documents – can be found at: http://www.lib.uchicago.edu/staffweb/depts/ils/kuali/index.html.

Data Migration and Integration Issues

For bibliographic and holdings data migration, the fact that the library was coming from a non-Unicode system required additional work. Kuali OLE also introduced a new type of holding called an EHolding record. Over time the library had dealt with electronic books, journals and other resources in a variety of ways. In some cases there were 865 fields in the MARC records, and in others there were also copy or item records. When we went to represent these materials in VuFind, it was decided to take advantage of the migration to OLE to regularize how we represented these materials by always using an EHolding record in OLE.

There were some special problems in data conversion due to the fact that we were moving from two separate systems, Horizon and III. These required custom data conversion scripts developed by in-house programmers.

MARC Authorities records were not planned to be supported in Kuali OLE until after the implementation of the early adopter sites. For a number of years, the library had sent new bibliographic records to Backstage Library Works and received corrected bibliographic and new and corrected authority headings. It was necessary to adapt this practice to continue to receive corrections for bibliographic records for loading into Kuali OLE, but to load authority records only into VuFind where they are used for cross references in the browse indexes.

Certain other features were not planned to be part of the initial Kuali OLE release and required that other local custom helper applications be developed. So, for instance, the Library used the spine label printing in Horizon, although not all libraries use that feature of an ILS. Also tracking payments accepted at circulation desks had some limited support in Horizon that was not in the new system, so some supplemental helper applications were required to support local workflows. The Library had developed many customized reports by pointing MS Access databases at the Sybase Relational Database Management System (RDBMS) of the Horizon system. Gradual adaptation of these reports to the Kuali OLE RDBMS is expected to take some work by the staff involved.

In addition, there are critical integrations required for the system to be functional. Specific Kuali OLE application programming interfaces (APIs) were developed to allow integration with the Dematic Automated Storage Retrieval System used in the new Mansueto Library. It was necessary to contract with the vendor for them to rewrite their side of that integration. The Library had moved Course Reserve functions to the Atlas ARES system and written a custom integration for the "place on reserve" and "remove from reserve" functions. Kuali OLE docstore APIs will be used to replace the custom integration at the RDBMS level used for Horizon. We are working with Atlas to implement this connection. NISO Circulation Interchange Protocol (NCIP) messaging to support participation in the UBorrow and Borrow Direct projects were also necessary parts of the project and require testing with the vendor systems. Finally, the ability to continue to extract payment information and format it correctly to load to the University Comptroller's system in order to pay our vendors was another critical integration that is required for going into production.

It was particularly critical that the VuFind Horizon connector be replaced by an OLE Connector in order to have the public catalog work correctly and provide the My Account features to allow self-service requesting and renewals. Because Villanova, where the primary VuFind development is done, was a Kuali OLE partner, they provided the OLE Connector based on the OLE APIs developed for this purpose.

A Basecamp project was used to track work on these various projects and a snapshot of that project is provided in Appendix C.

Technical Migration Activities

Staffing

The Integrated Library Systems group included a manager, Library Systems Analyst, Systems Librarian, Database Administrator, Senior Programmer Analyst and Library Operations Assistant, who also did some web programming. To accommodate the need to support the open source systems, another entry level Programmer Analyst position was added to help support VuFind and a Senior Programmer Analyst with Java skills was hired to help support Kuali OLE. This staff would support the migration, as well as the integrations with other systems and the add-on custom applications needed for optimal use of the system. In addition, web programming staff and the Web Program Director for the Digital Development Library Center were used in the VuFind project to customize that system.

Hardware/Software

While a number of the commercial library systems are moving to "cloud-based" systems, there was no real impetus to consider such a solution at UChicago. Indeed, in a university setting there can be obstacles to such an implementation. For instance, during the course of the project, Lehigh University – another Kuali OLE partner – made a decision that university financial data should not reside in the cloud. Chicago had not made any such general policy, but major system implementations do require security and architecture reviews. Patron database information in particular would be problematic to be made available on a commercial, vended system. VuFind and OLE will be implemented on virtual servers hosted by university computing. In the future, if the university offers cloud-based hosting services, it will be possible to take advantage of that. OLE itself is being developed on equipment that is in the Amazon cloud, so it is demonstrated that it can be run in that environment. At the moment, universities have some legal reservations about agreements to run on cloud-based commercial systems. It was seen as reasonable for the Library to follow university policies in this area and not to attempt to run the library systems separately. The intention is to take advantage of the university enterprise systems for storage, backup and system administration.

Plans for migration of the library system will undergo a review by the university ITS Technical Architecture Committee and also a security review. Appendix D contains the representative list of questions for these reviews. A separate PDF is attached which contains the diagram that the Library provided for that review.

A basic difference in implementing open source software is the need to pull down source code to a development environment and to develop a process to deploy new versions and fixes. This is true for both VuFind and Kuali OLE. This required some upgrades of equipment in the library for development and testing before deployment to production environments in the university data center.

Post-Migration Planning/Sustainability

Based on previous experience, the Library is well aware of the possibility of system instability during the first few weeks of production. Library staff will be forewarned of this; contingency plans allow the Library to resume using Horizon or Millennium if that should become necessary. "Hope for the best and plan for the worst" is a prudent guiding principle for any new software implementation.

During the first few months of production, Library staff will undoubtedly find bugs and discover gaps in our training. ILS staff will be prepared to spend most of its time with intensive troubleshooting.

The Kuali OLE software development schedule calls for an implementable version to be completed in late spring 2014 and for patches to be available during the summer of 2014 for the two early implementer libraries. Experience of the other Kuali projects has been that early implementers developed a number of bug fixes and customizations for local implementations that were difficult to eventually include in the codebase. As a result, the OLE project will plan to support and merge the patches for the early implementers into the 2015 version of the software that will include additional functionality desired by all of the partner libraries. Implementation at the University of Chicago is planned with the intention of continuing to participate in development and testing of this next version. For the first year, the approach for local functionality to supplement Kuali OLE features will rely on helper applications separate from OLE, rather than modifications to the code itself.

Procedures for code contributions are being developed by the Kuali OLE Technical Council, and it is anticipated that there will be ways to do this by the time more partners implement in 2015. Appendix E contains the draft of these procedures. These processes are already in place for VuFind software and the library has contributed code where appropriate during the customization of VuFind for local use. While it has always been the plan to customize VuFind for local usage, early and intense involvement with the OLE development has resulted in a strategy to rely on local customizations as little as possible in the ILS implementation.

Our training plan calls for staff meetings about a month after the initial production date, organized by functional areas, to provide an opportunity for training refreshers and discussion of any ongoing issues. Additional follow-up meetings may be required, depending on the volume of issues.

Once Kuali OLE is reasonably stable, ILS staff will need to migrate seven years' worth of past acquisitions data from Millennium to a database of some kind. Horizon historical circulation data will also be migrated to a data warehouse to support reporting. The intention is to reactivate a project to implement a data warehouse and a more sophisticated system for reports and analytics in the year following implementation of Kuali OLE.

During the six months after implementation, any remaining useful data will be moved from Horizon and Millennium. Those systems will cease to be updated at the time of cutover. They will be available for consulting until they are retired in January 2015. HIP and AquaBrowser will be unavailable once the cutover to OLE happens, as they do not point to OLE. They will be retired as soon as the OLE system is stable.

Lessons Learned

It probably goes without saying that any system migration is a difficult and time-consuming process for all library staff. It is even more so when the new system is still in active development, which of course complicates matters. We would certainly advise any library to wait for a completed product before attempting a migration unless: (1) you feel confident that your staff can handle the situation of being beta testers at the same time; AND (2) there are compelling reasons – financial, functional, or other – to move off your present system ASAP. Both of these applied to the University of Chicago.

Commitment to developing a community-sourced project such as Kuali OLE requires a library to take a hard look at its staffing levels and the available skill sets. The ability to draft functional requirements, write test scripts and perform Quality Assurance/Quality Control (QA/QC) work, and write coherent documentation are necessary for any software development project and are not necessarily widespread among library staff. There is sometimes also a perception that this work must come "after" other duties – which it cannot if development schedules are to be met. Libraries used to complaining about the inadequacies of their vendors need to realize that with community-source developed products, what was formerly "them" is now "us", and whatever inadequacies exist can only be traced back to our own doorsteps. In other words, a commitment to develop software cannot be taken lightly. At least as Kuali OLE is financed, these efforts depend on library staff, not paid employees as is the case with commercial vendors.

We anticipated that running open source would require more technical resources; we hired two additional programmers, and we believe that this turned out to be a wise move on our part. We could not possibly have gotten as far as we have with either our VuFind or Kuali OLE implementation without these additional resources. Even if we had decided to run open source without being a development partner, we would still have needed at least one more programmer.

Because of the active development, it turned out to be impractical to contract out for data conversion and training. If we had waited for software development to be complete it probably would have been cost effective to contract out some of that work. Also, the timing of internal work for the project was approximately a full year. There were many iterations of data conversions and installation and setup because we were not working with the final, stable version of the software. Implementation probably could have taken half the time if not for this situation. On the other hand, it was necessary to forge ahead with figuring out setup issues and working on data cleanup and conversion issues in order to meet the desired schedule.

Work with the Kuali project has been useful in forging alliances with our university computing groups. The Library has benefited by association with an open source project that is contributing to other areas of academic computing, particularly in the area of identity management.

Appendices

These appendices can be found on the FOSS4Lib.org website:

- UChicago's OLE-deployment-diagram.pdf
- APPENDIX A. Kuali OLE staff contributions
- APPENDIX B. Training Plan
- APPENDIX C. Basecamp project snapshot April 2014
- APPENDIX D. Questions from Technical Architecture Review Committee
- APPENDIX E: OLE Contribution Requirements