

# **No more lines! Patron Update Web Services at the University of Lethbridge.**

Leona Jacobs  
University of Lethbridge, Library  
and  
David Howell  
University of Lethbridge, Information Technology

Presentation to the 2009 Innovative Users Group held May 17-20 in Anaheim, California  
2009 May 18 and May 20

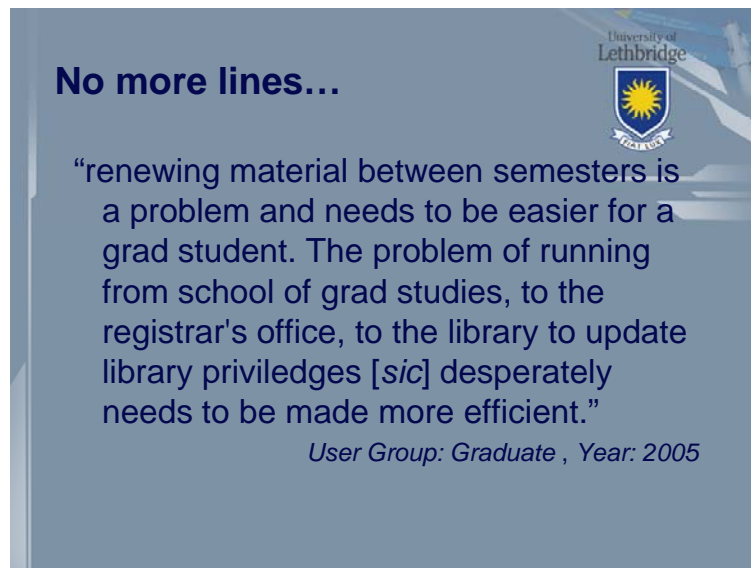


Thank you for coming. This session is titled, “No more lines: Patron update web services at the University of Lethbridge.” My name is Leona Jacobs and, I guess, I am the program convener since I am the one who submitted the proposal! I have been a librarian at the University of Lethbridge in Lethbridge, Alberta, Canada for the best part of 20 years.

David Howell has had a peripatetic career that has taken him to places near and far. He currently works as a Programmer/Analyst in the Enterprise Information Systems unit of University’s Information Technology department. He has worked for the University almost 18 months, mostly on this project.

Rice Majors is a product manager at ILL and was assigned to Web Services at the time that we were trying to find our feet with this project. He was at the other end of email or the phone encouraging and facilitating our

implementation of this product. It is only right that he sits with us today as a member of our team and he is willing respond to any questions that come up about the Patron Update Web Services specifically and III's development of APIs in general.



The title, "No More Lines" is derived from a directive from University Administration – which did not dictate but did coincide with this project – the idea being that as a University, we needed to streamline our processes and make the student's first experience, that of New Student Orientation, a positive one. The intended beneficiaries were first year students at the undergraduate level, but as you can see from this slide, we had and have work to do at the graduate level as well.

In the summer of 2007, the University of Lethbridge Library invested (and I use that term on purpose) in one of III's rare deals billed as the "Campus Integration Suite." This package of software included, among other things, the Patron Update Web Services and the Fines Payment Web Services. And so our tale begins.



Ultimately, the implementation of these two pieces of software has broken down into four parts:

- Part 1 was to develop the means of updating existing patron records for students, faculty, and staff. In addition, it included creating new student records as required. This is done and functioning.
- Part 2 consisted of translating the student's registration into library privileges. This piece is done and functioning.
- Part 3, as planned, is to add in the creation of patron records for faculty and staff and updating of library privileges (including, the addition of notification of retirements and ultimately death).
- Part 4, as planned is to create the ability for any fines or fees to be applied to the student's financial record with the Cash Office with a posting of the Library's line item directly back into the patron record when the fines or fees have been paid (i.e., no more phone calls back and forth).



## LBCIP Goals ...


- Real-time
- No staff involvement

*“...an aggressive implementation”*

-- Rice Majors, 2008-08-18

We want all of this to happen in real-time (or as real-time, as we can get it) and directly system-to-system (i.e., no or minimal staff time involved in processing).

Upon completion, Rice deemed that this was an aggressive implementation which he tells me, was “speak” for “oh my god, they did it!”



## LBCIP (student) key dates ...

- 2008 January 22:
  - work begins in earnest.
- 2008 July 8
  - Demographic is turned on
- 2008 August 15
  - first reconciliation of Banner registrations

Part 1 and Part 2 combined took a concerted effort and many meetings but once we got our heads wrapped around the concept, we were up with our real-time student registration in time for our Fall semester.



Today, I am going to introduce you to the University of Lethbridge, tell you the tale of our cranky students, and layout for you, the pieces of this project that I was most involved in. David is going to outline the technical side of the project and then I will do a bit of a wrap-up. Our goal is that you leave here appreciating the impact that web services can have on your library and on your university community.

That said, I want to preface my remarks by saying that we were very naïve about what we were embarking on. You won't be.

**The University of Lethbridge and Library**

The University of Lethbridge is, by Canadian standards, a small-on-the-cusp-of-medium university located in Lethbridge, Alberta, Canada. So where the heck is Lethbridge?!

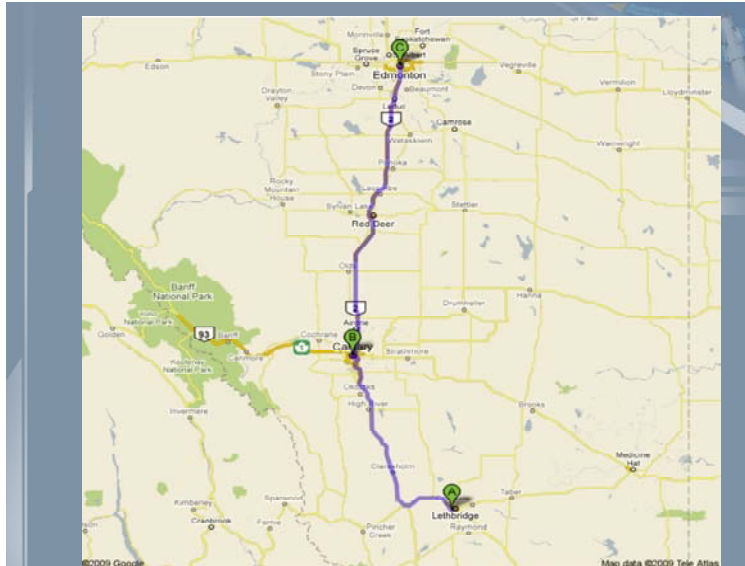


Well , assuming that you know where both Canada and Alberta is, Lethbridge is a small city (~ 84,000 people in 2008) ... situated in the deep south of the province of Alberta about 1 hour north of the Canada/USA border and about 3 hours north of Great Falls, MT.

## Lethbridge (Main) campus ...



The University of Lethbridge is situated on the west side of the Oldman River which runs through Lethbridge. There are three campuses serving approximately 8000 students at the undergraduate and graduate levels. Although we have always had a well developed Masters of Education program, the University has until recently described itself as a “primarily undergraduate, liberal education” institution. Recently, however, changes in the post-secondary education system in Alberta have resulted in the UofL being designated as a “comprehensive” university focused on graduate studies and research so instead of drifting towards graduate studies, we are making a sharp turn towards building our graduate studies program. This is important because there is now a lot of attention being paid to the issues raised by graduate students.



Besides the main campus in Lethbridge, which serves about 7100 of the students, there are two “northern” campuses in Calgary and Edmonton, ~250km and ~500km to the north of Lethbridge, respectively. As of Fall 2008, approximately 450 students were enrolled through the Calgary campus while approximately 350 were enrolled through the Edmonton Campus. There are no branches of the Library set up in either Calgary or Edmonton. Instead, these users take advantage of our fairly extensive offering of electronic resources, our delivery of physical materials on extended loans, and a province-wide reciprocal borrowing program will facilitates their use of libraries closest to where they live, whether those libraries are public or academic. One important bit is that the access to the electronic collection, on which they have a high dependency, is authenticated through our Web Access Management module of Millennium so they require current library privileges.

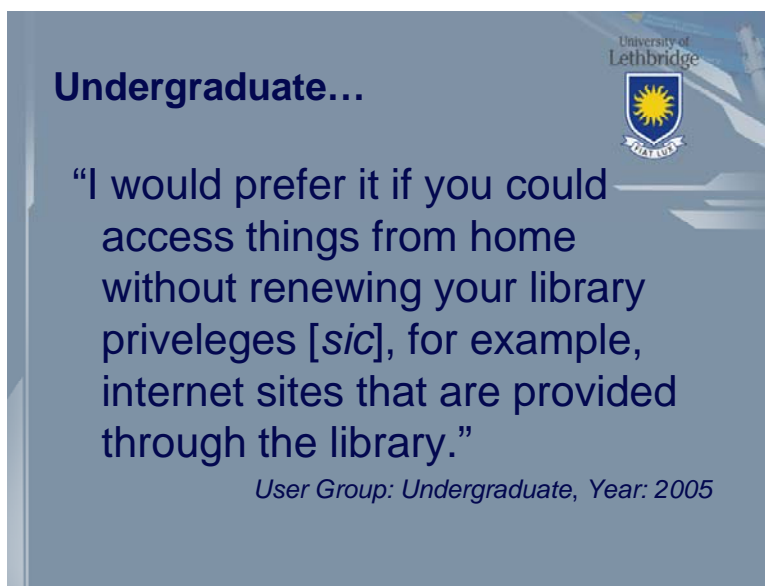
***A bit about how we (used to) do things, or, the case of the cranky students***

When we implemented Innopac in 1989-90, we did what I suspect most libraries did – an initial load of patron records based on the output from some other system. Once that initial load was done, then the patron records were maintained manually, on a semester-by-semester basis. Prior to the start of each semester, we would extract and load from the Registrar’s system the records for newly admitted students only. These records were blocked from use and the records of returning students were not updated until such time as the student showed up at our circulation counter with proof of course registration, often when they wanted to finally borrow something from the Library. Then we would theoretically update their contact information and definitely their library privileges, check out their materials, and send them on their way. As I am sure you can relate, September and January, the start of our main academic semesters, were a nightmare of line-ups (which grew as the campus grew) and we often had two staff or more updating cards during this period.



Although at a distance, the same process was applied to assigning library privileges for students at the Calgary and Edmonton. In this case, the in-person presentation was replaced by an online “application form” which had a stated turnaround time of 24-hours – during the week – although some cross-training of weekend staff helped facilitate weekend applications.

This seemed to be working just fine ... at least from our perspective. However, when we ran our first LibQUAL+™ survey in 2005, we discovered we had some rather cranky students on our case. To illustrate, the undergraduate response rate was 28% while for graduate students, it was 33.9. If you have ever done LibQUAL+™ or some other web-based survey with a group other than librarians (who tend to be very enthusiastic), you will appreciate that the response rate alone was a comment on the Library. In other words, they had a few things to tell us but in particular, that Library needed to wake up to two things:



- 1) The Library had expanded the electronic offerings which required students to have updated library privileges in order to access these resources and students, who had not taken the time to get their privileges updated, were finding themselves locked out of their online library.

## Graduate students ...

University of  
Lethbridge



“I find one HUGE problem for me is that at the end of every semester, the library privileges must be updated. I cannot do this until I'm registered for the next semester, which I can't do until I have my committee meeting. Very often I am without library privileges because I can't register for the next semester yet.”

*User Group: Graduate , Year: 2005*

2) the Library procedures weren't working for graduate students.

You pick your battles and post-LibQUAL+™, while I took a sabbatical leave, the then Associate University Librarian took over responding to the survey. One of the battles that she picked was to improve on our services to graduate students starting with the procedures for registering them. However, after months of meetings and emails discussing various, very complicated options, all of which were labour intensive and exclusive to graduate students, she was no further ahead. I returned to work around the same time as we became aware of the Patron Update Web Services (Summer 2007).

## Sahai & Graupner (2005)




**“Web services provide a front-end for the internal business processes of an enterprise. They help expose subsets of the internal business processes and activities so that external business processes may compose with them. This leads to what is termed an orchestration or choreography. The enterprise business processes have to be defined and some of their activities have to be linked to the WSDL [web services description language] operations. This requires modeling of Web service's back-end business processes.**

What are web services? Sahai and Graupner in their book, *Web Services in the Enterprise* (2005), describe web services as follows – the emphasis is mine:

**Web services provide a front-end for the internal business processes of an enterprise. They help expose subsets of the internal business processes and activities so that external business processes may compose with them.** This leads to what is termed an orchestration or choreography. The **enterprise business processes have to be defined** and some of their activities have to be linked to the WSDL [web services description language] operations. **This requires modeling of Web service's back-end business processes.**

We will work through each of these pieces individually.




**Sahai & Graupner (2005)**

**“Web services provide a front-end for the internal business processes of an enterprise.** They help expose subsets of the internal business processes and activities so that external business processes may compose with them. This leads to what is termed an orchestration or choreography. The enterprise business processes have to be defined and some of their activities have to be linked to the WSDL [web services description language] operations. This requires modeling of Web service's back-end business processes.”

**“Web services provide a front-end for the internal business processes of an enterprise.”** Patron Update Web Services, in particular, is the means by which a patron record is searched, created, updated, and deleted using a key common to both the internal system (Millennium in this case) and the external system (for example, Banner). In our case, the common key was the university identification number or UID.

If web services is as foreign a concept to you as it was to me back in 2007, I encourage you to attend the Innovative Presentations on APIs (K10 and P11). Although we can look back today and say that implementing PUWS, in the way we have isn't technically difficult, at the time we felt that simply acquiring this piece of software was the easiest part of the process.



**Sahai & Graupner (2005)**

“Web services provide a front-end for the internal business processes of an enterprise. **They help expose subsets of the internal business processes and activities so that external business processes may compose with them.** This leads to what is termed an orchestration or choreography. The enterprise business processes have to be defined and some of their activities have to be linked to the WSDL [web services description language] operations. This requires modeling of Web service's back-end business processes.”

***“They help expose subsets of the internal business processes and activities so that external business processes may compose with them.”*** The “external business process” that we wanted to collaborate with was, in this case, the student registration process in the Registrar’s Office. In return, the Registrar’s Office saw a couple of benefits for their operations:

- 1) the “carrot” of library privileges, hence access to electronic resources, in exchange for prompt registration in the case of graduate students. As I have mentioned, the internal workflow getting graduate students registered was a tad “casual” which not only was frustrating to the students but, as it turned out, frustrating to the Registrar’s Office.
- 2) By giving up maintaining student patron records and instructing students to maintain their own contact information in Banner, the Registrar’s system would become the authorized source for all contact information about the student. No more second guessing whether the Library’s information was more current.

Here in lies the key. Communicate. Find common ground. Appeal to the higher ideal if you have to (e.g., common philosophy, student experience, etc.)

Although the Registrar’s Office had other “irons in the fire” so to speak – they were involved in complying with the development of a provincial-wide application process – they wanted to move forward on this project. Now the Library had no programmers to make this happen but the Registrar’s Office did. However, given the lack of knowledge about what was entailed, they were uncertain whether and how they could commit. Thus the Fall of 2008 was spent trying to determine what was involved. The idea was that if, we could find a Banner site that had already done what we wanted to do, we might be able to borrow and adapt existing code or at the very least, draw on the experience of others in considering our own project.

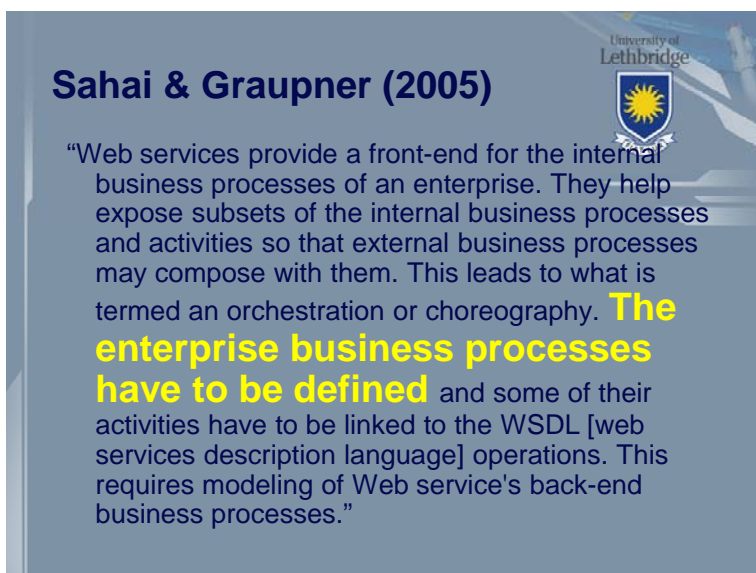
To this end, I mined all the all the contacts that III provided to me. I searched the IUG list archives for possible contacts. I followed leads from other sites. I grew appreciative of the various ASK US services out there when it wasn’t obvious who I should be talking to. Thank you to all of you for the time spent emailing or phoning. ☺ As it turned out, there were a number of III customers who had purchased this product but either:

- They weren’t a Banner site – PeopleSoft being the usual alternative it seemed. And in these cases, they either hadn’t developed anything yet or were using in a way that didn’t fit our vision ; OR
- They had the product but didn’t have the resources to move ahead themselves and couldn’t get buy-in from their IT folks to build something for them.

My Banner counterparts in the Registrar’s Office tried from their side and found the same thing: lots of Banner-Millennium sites who were in the same boat as us and wondering how we were approaching the situation. The

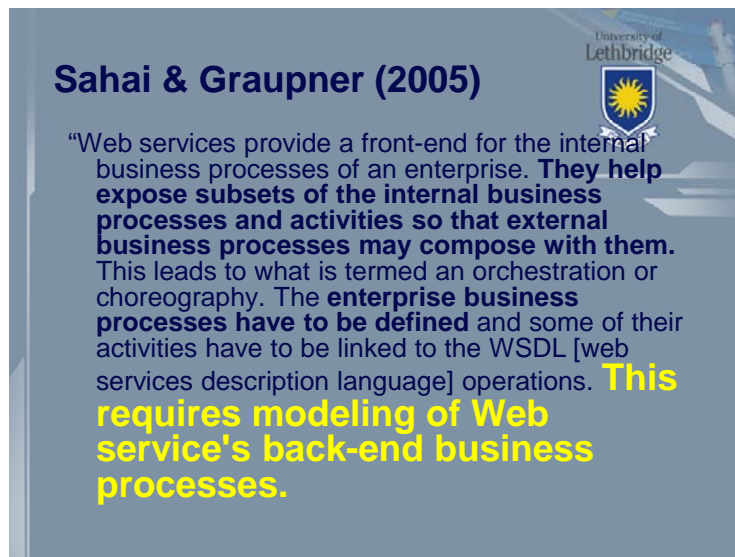
date by this point was December 2007; we had spent somewhere between four and six months trying to find an easy solution.

Given that we found no site with experience to draw from, the Registrar's Office contacted SunGuard, the vendor for Banner to see if they would be willing to consult. On January 22, 2008, we had a conference call between representatives of SunGard, Doug Randall and Rice Majors of III, the people involved from the Registrar's Office, representatives from the newly formed Enterprise Information Systems Unit (including David) who had been assigned overarching responsibility for Banner, my Associate University Librarian and me. SunGard was willing to help us define our business processes ... for a price – a price predicated on the desperation we were feeling. The Enterprise Systems folks said thanks but no thanks and when the call was ended, expressed their desire to build it in-house and David, in particular, felt it wouldn't be that hard of a project. The Registrar's Office people looked relieved. The Library people were thrilled. A project team was struck.



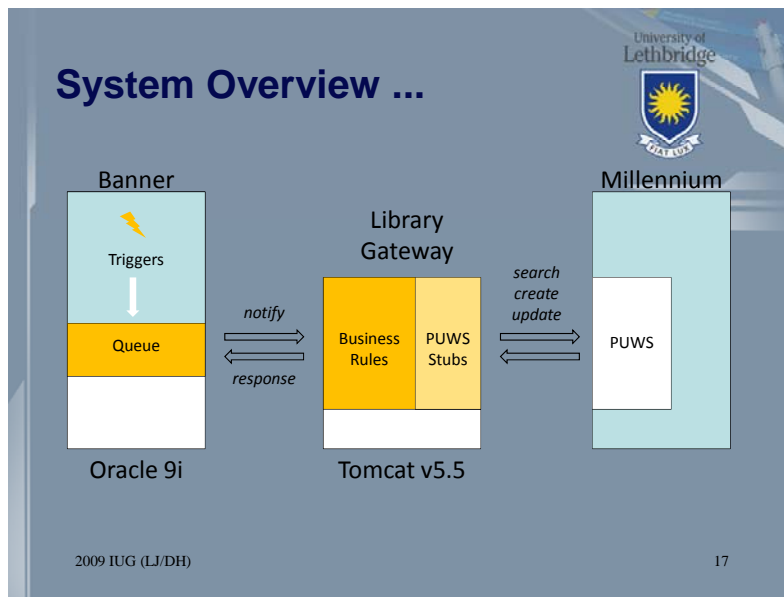
**“The enterprise business processes have to be defined.”** Take my advice. Do not attempt anything relating to developing web services unless and until you have all the relevant players in the room to talk about needs and processes. It is tedious. It gets tense. It takes tenacity. It is worth every minute because it forces both parties, as long as you have found that common ground (key point!), to confront the *status quo*. Within the Library, we challenged the patron record data we had been collecting and forced a re-justification for collecting each piece; if there was a good case made, then it stayed, otherwise it went. That said, you need a mixture of people at the table from those parts of the organization affected: front-line staff who know the workflow, technical staff who know the systems involved, and, finally, those with the authority to compromise where and when compromise is called for. Everything is documented: the process required, where the data was going to come from, where it was going to be mapped to, when, how, etc.

From the end of January to the end of March, essentially eight weeks, two teams of people met almost, if not, weekly: those involved in defining and documenting the overall project and those involved in defining and documenting the business processes relating to creating and updating a patron record. I have a copy of our Business Requirements Document for this project here if anyone wants to see it. We have also negotiated an addendum to bring in faculty and staff data from Human Resources and to illustrate the benefit of this type of effort, David and I were discussing the HR piece and he thought we still had some unresolved issues but I was able to point the negotiated solution in the BRD.



Finally, **“This requires modeling of Web service’s back-end business processes.”**

At this point, I am going to turn the presentation over to David.



When students add or drop courses or change their personal information:

- triggers in the Oracle database add entries in a table-based queue in the database.
- queue entries are then forwarded to the Gateway.
- the Gateway handles the Library business rules needed to map data values, determine expiration dates, etc before updating the patron record via PUWS.

The Gateway searches for, creates and updates patron records via PUWS using generated stub classes to interact with PUWS. The Gateway returns OK/RETRY to the queue manager (more about this later).

Students are granted lending privileges on a per-term basis, according their course registrations. They get privileges as soon as they register for their first course. If they drop their last course, their privileges are terminated.



## Requirements and Constraints



- Live updates to patron records in Millennium.
- Minimize manual intervention by IT and library staff.
- Robust handling of unavailability of system components.
- No degradation to experience of Banner users.
- Start of transition from PL/SQL to Java, Web services.

2009 IUG (LJ/DH)

18

Primary goal was to provide live updates to Millennium when student information in Banner changes.

Some batch operations are still needed, but this goal has been achieved – Millennium is updated 24/7 within several seconds of the change in Banner.

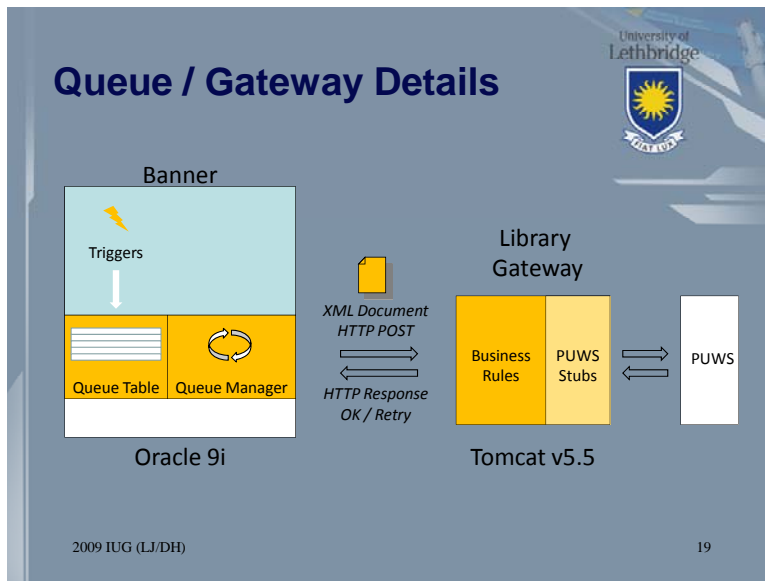
Didn't want IT or Library staff to have to run exports, manage synchronization with Millennium backups, etc.

Implies the system needs to cope with network outages and with situations where component systems such as Millennium are off-line for backups or maintenance. We wanted to avoid having to know about scheduled backups, and didn't want to have to do significant manual tidying up after ad-hoc backups or other incidents.

Also, even when the system is up, the PUWS web service operations can return results such as a record being busy or stale, and you need to be able to handle that gracefully.

The process of moving an update through the system involves a couple of hops and a couple of PUWS operations, so takes about 3 seconds. We don't want the Banner user waiting for this, or to be affected by other system issues, so needed to find some way of decoupling the different parts of the system.

The team is comprised of Analyst Programmers who know the Oracle database and PL/SQL, but are making the transition to more use of Java, Web services and service orientation. Some of the implementation decisions that we took reflect this.



The system needs to process updates from Banner asynchronously (without making the Banner user wait for Millennium to be updated) and it needs to reliably deliver the update.

We met both of these requirements by implementing a simple, table-based queue in the Oracle database. Could have been done using a message broker (JMS) but at the time we had more Oracle skills and this represented a lower-risk option.


Changes to Banner are captured by triggers we added to several Banner tables. The only thing the trigger code does is to insert an ID into the queue table.

A separate Queue Manager job polls for queue entries that need to be processed, assembles the XML that the Gateway requires and forwards it to the Gateway. The Gateway is a Java servlet; the queue manager just does an HTTP POST.

We also implemented a retry mechanism. If the update can be forwarded to the Gateway and the Gateway can successfully update Millennium, the Gateway returns an OK response to the Queue Manager and the update is removed from the queue. If the Gateway can't be contacted or for some reason can't successfully process the update, it responds with a request to retry the update later.

Updates are retried every 10 seconds for an hour. If undeliverable at that point they are marked as dead entries, but are retained in the queue.

This approach successfully handles a wide range of outages: the Gateway being off-line while being upgraded or restarted, Millennium backups and upgrades, etc.



## The B Word ...


- Some batch operations are needed.
- Bulk reconciliation is run prior to each term.
  - Advance course registrations are revisited.
- Use Oracle stored procedure to add entries to queue table.
- Manually initiated (for now).

2009 IUG (LJ/DH) 20

The system processes demographic updates live throughout the year, but registration updates only become valid once the relevant term has started.

Advance registration updates are filtered out by the Gateway. Just before the start of each term we run a bulk reconciliation: a SQL query identifies the students registered for the upcoming term and their IDs are inserted into the Oracle queue table for processing. The Gateway will at this time be treating the upcoming term as the current one, so will grant the appropriate privileges for the term that is about to start.

We can run about 8000 students through the system overnight. Each update takes about 3 seconds to process. This could be optimized, but so far hasn't been an issue.



## Lessons Learned ...

- Using PUWS is not complicated.
- System complexity is elsewhere.
  - Business rules: “What is a student?”
  - Mechanism for asynchronous, reliable delivery.
  - System operation needs to be monitored.
  - Design, code quality for 24/7 operation.

2009 IUG (LJ/DH) 21


The PUWS Web services themselves are relatively straightforward to use. Using Java, C# or other languages that have good tools for SOAP Web services, you can generate client stubs and be interacting with Millennium very quickly.

However, the PUWS Web services only support basic operations on patron records – create, search, update, delete. Most of the project time was spent on other areas:

Discovering and documenting business rules, determining where information about student status is kept.

Handling requirements for asynchronous, reliable delivery of updates.

Monitoring the operation of the system has been important: need to get notifications when problems occur.



## Moving Forward ...

- PUWS wish list:
  - Better handling of status during backups.
  - Events from PUWS when fines are incurred.
  - Use document/literal SOAP style.
- Our system:
  - Automate per-term reconciliations.
  - Improve the queue implementation.
  - Transfer library fines to Cash Office accounts.
  - Handle staff and faculty updates.

2009 IUG (LJ/DH) 22

We've found the PUWS Web services to be very useful, but there are a couple of areas where they could be improved.

The current PUWS implementation is a SOAP Web service that uses the rpc/encoded style. We've had problems using this with more recent Java SOAP implementations and rpc/encoded is generally a deprecated style of Web services.

Currently, when a Millennium backup is running the PUWS Web services operations can behave in ways that makes it difficult for a client to know what is happening in Millennium. createPatron() results in a pending database operation and multiple client records can result. Just returning busy, would make it easy for us to manage the system.

We would like to have a similar, event-based system for transferring library fines to the student's account with the Cash Office, but currently don't have an interface for detecting these events.

Are revising the queue manager implementation: support multiple subscribers and move the main polling loop out of the Oracle database into a Java component.

**Conclusion:**



Alas, we ran our second LibQUAL+™ survey in the spring of 2008 while we were still developing the interface between Banner and Millennium. However, it is hard to keep good news a secret and this comment came in as part of that survey.

When September rolled around, a number of students came up to the desk to get their library privileges updated only to be told that they were good to go. They were both surprised and pleased. Others came up to the desk to confirm what they had heard second-hand from friends. When staff confirmed the rumour, they were both surprised and pleased. We heard a lot of “sweet!”

Typically, in September and January, two staff and a student were scheduled to be at the circulation desk during the regular hours. While we have not yet dispensed with that scheduling, the staff quickly decided to split those shifts as the Gateway was now doing most of what had been their work during this period. One skeptical staff member, who, as per instruction, had sent a student to Banner to update their contact information, was amazed when the student came back 5 minutes later and the updated information was already posted to Millennium.

I have become a fan of the concept of Web Services and the potential this technology holds for streamlining operations and extending the presence of the library into other areas of the University. In addition to the Patron update Web Services and the Fines Payment Web Services, my library has become a “development partner” for the MyMillennium Web Services and anticipate its use in the upcoming employee and student portals. None of the III products we have is 100% perfect but to be generous in this criticism, I will attribute this to its recent emergence and evolution as we put it to work for us. Developmental bumps aside, I am very much of the belief that III is on the right path by developing the APIs that expose a library’s data for use with and by other

administrative systems. I agree with John Blyberg of Darien University in Connecticut (2007) who, in writing about integrated library systems of the 21<sup>st</sup> century, has called for, among other things, “a full-blown, W3C [World Wide Web Consortium] standards-based API to all read-write functions.” Bring them on.



## Reference List

- Blyberg, J. (2007). Mouse bites cat: taking back the 21st-Century ILS. In R. S. Gordon (Ed.), *Information tomorrow : reflections on technology and the future of public and academic libraries* (pp. 31-41). Medford, N.J.: Information Today, Inc.
- Sahai, A., & Graupner, S. (2005). *Web Services in the Enterprise: Concepts, Standards, Solutions, and Management. Network and Systems Management*: Springer.
-