Extending Reports into Services: New Infrastructure, Tools and Contexts

Kevin Kidd
Library Applications & Systems Manager
Boston College
University Libraries

ELUNA 2010
Fort Worth, TX
May 13, 2010
2:15 – 3:15 PM
Agenda

Ø Web Services and Library Services: Our Strategy and Requirements

Ø Administrative (e.g. Back Office) Reporting Needs

Ø Online Patron Services: (e.g. Customer-Facing) Needs

Ø Our Solution: LogiInfo and LogiAd Hoc

Ø What’s Next: Future Developments
Some Background...

Ø By Early 2008, the BC Libraries were in desperate need of a new Administrative Reporting System

Ø Legacy Crystal Reports System was woefully inadequate (8 Years Old)

Ø We had recently developed a strategic plan (the Aerie Project) to build a flexible web services infrastructure to deliver new patron and back-office services

Ø The goal of the Aerie Project was to create online library services that could be combined and reused in different contexts

Ø Library web site
Ø Blackboard CMS
Ø University administrative portal
Ø Personal research portal
Ø Personal Presence Web (share.bc.edu)
Initially, we approached the problem on two separate tracks

Ø Researched and tested multiple reporting and BI applications (Open Source and Commercial)
  Ø Jasper
  Ø Pentaho
  Ø ARC
  Ø Crystal Enterprise
  Ø LogiReports / LogiInfo

Ø Simultaneously began education and research on web services platforms and solutions
  Ø Began to build in-house Java programming expertise.
  Ø Mule ESB
  Ø JBoss ESB
  Ø Oracle ESB
Features we were looking for in a reporting solution

Ø Ease of Use; Intuitive Interface

Ø Ease of Deployment (Web Browser Only; No Plugins)

Ø Scheduling of Reports

Ø Open Environment (e.g. Linux/Tomcat/Apache, etc…)

Ø Ability to integrate data from multiple sources (Aleph, SFX, MetaLib, X-Services, etc…)

Ø Reports available in multiple formats (HTML, PDF, Excel, CSV)

Ø Viability – lots of users; active developer community

Ø Reasonable license cost and terms
Features we were looking for in a web services / online patron services platform

Ø Here, our initial sense was what we required was, admittedly, vague

Ø We wanted to be able to “glue” together unrelated data and services to build and deploy new services.

   Ø X-Services
   Ø Aleph, SFX, MetaLib Data
   Ø ILLiad and other applications

Ø Reusability of these new services was very important

Ø Open standards are very important as well (Java, Tomcat/JBoss)
With all of this in mind, we initially settled on

LogiInfo BI Platform

Why?

Ø Purely web-based reports - no plugins or additional software

Ø LogiStudio: Easy to learn report development environment - no coding required

Ø Support for a Java application servers – BC uses Tomcat

Ø Access and join data from virtually any data source – SQL, Web Services, Text Files, XML, Google, Twitter, Screen Scrape, Excel, etc…

Ø Export to Excel, Word, PDF, CSV, HTML, and XML
But, the clincher was

Server-based, NOT user-based, licensing

Why is this crucial?

◊ Much more economical. Don’t need to pay for individual users.

◊ LogiInfo potentially becomes a full-blown dynamic web application development environment, not simply a back-office reporting system.

◊ We are free to deploy applications for everyone at Boston College. Not just library staff.
We realized that LogiInfo could actually fulfill many of the web service needs we identified in planning for the Aerie Project.

For example, LogiInfo’s ability easily to “glue” together data from multiple, unrelated sources would allow us to build a context-sensitive Course Reserve service that:

Gets a Student’s Current Schedule (Source: BC Scheduling System)

+ 

Gets the items on reserve for a given course (Source: Aleph DB)

= 

New Service: My Course Reserves
Also, because this new My Course Reserves service was built in LogiInfo, it is highly portable - another need we identified in the Aerie plan

Ø For example, we can embed the service in Blackboard/Vista CMS, in the University Admin Portal and in course-specific LibGuides.

Ø The service can be inserted directly into an HTML page via a JavaScript “Widget”.

Ø Styles (CSS) can change on the fly to seamlessly integrate with the host web page or application.

Ø The service can mimic Aleph X-Services, and return the course reserve data as pure XML that can be used in any way the calling application requires.
Sample Widget – To embed in HTML. Automatically generated by LogiInfo

```html
<FORM> <!-- The FORM element is required if there are INPUT elements in the Widget -->

<!-- Include this once for each page. -->
<!-- Include this for each widget on the page. -->

<SCRIPT src="https://arc.bc.edu:8443/LibraryAdminServices/rdTemplate/rdWidget/rdWidget.js">
</SCRIPT>

<!-- Include this once for each page. -->
<DIV ID="myWidgetDiv" >Loading... </DIV>

<SCRIPT>
  var myWidget= new rdLogiWidget;
  myWidget.definition="CourseDocs";
  myWidget.containerID="myWidgetDiv";
  myWidget.setParameter("lgxPreview","41403");
  myWidget.load();
</SCRIPT>

</FORM>
```
Sample XML for Requested In-Process Web Service

<rdData>
<dtRequestedInProc PATRONID="0000012826" NAME="Ashley,Jane" EMAIL="ASHLEY@BC.EDU"
PHONE="6175524155" SYSNO="002671977" TITLE="Educating nurses : a call for radical transformation / Patricia Benner ... [et al.] ; foreword by Le"
AUTHOR="" BARCODE="B947846"
REQ_DATE="20100506" PICKUP_LOC="ONL"
REQ_ARRIVAL_DATE="20100222" calcReqDate="2010-05-06"
calcArrDate="2010-02-22" calcPickupLoc="O'Neill Library"/>

<dtRequestedInProc PATRONID="0000011418" NAME="Bregman,Adeane" EMAIL="BREGMAN@BC.EDU"
PHONE="6175523136" SYSNO="002678481" TITLE="Der Tod - Tor zum Leben ein Trostbuch Dietrich Steinwede"
AUTHOR="" BARCODE="B924574"
REQ_DATE="20100510" PICKUP_LOC="BAPST"
REQ_ARRIVAL_DATE="20100308" calcReqDate="2010-05-10"
calcArrDate="2010-03-08" calcPickupLoc="Bapst Art Library"/>

<dtRequestedInProc PATRONID="0000011418" NAME="Bregman,Adeane" EMAIL="BREGMAN@BC.EDU"
PHONE="6175523136" SYSNO="002683693" TITLE="The Hudson River to Niagara Falls : 19th-century American landscape paintings from the New-York Hist"
AUTHOR="" BARCODE="B927861"
REQ_DATE="20100511" PICKUP_LOC="BAPST"
REQ_ARRIVAL_DATE="20100325" calcReqDate="2010-05-11"
calcArrDate="2010-03-25" calcPickupLoc="Bapst Art Library"/>

<dtRequestedInProc PATRONID="0000012826" NAME="Ashley,Jane" EMAIL="ASHLEY@BC.EDU"
PHONE="6175524155" SYSNO="002671977" TITLE="Educating nurses : a call for radical transformation / Patricia Benner ... [et al.] ; foreword by Le"
AUTHOR="" BARCODE="B947846"
REQ_DATE="20100506" PICKUP_LOC="ONL"
REQ_ARRIVAL_DATE="20100510" calcReqDate="2010-05-06"
calcArrDate="2010-05-10" calcPickupLoc="O'Neill Library"/>
</rdData>
Requested In-Process Barcode Flags Generated from XML Service
What else can we do with LogiInfo?

Ø “My Library”-like portal (using Dashboards)

Ø Library Resource Recommender (based upon Major, a patron’s course schedule, or other user-selected criterion)

Ø Easily build interfaces to Aleph, MetaLib and Primo X-Services and/or SOAP Services, including view/renew items, search the catalog, etc

Ø Build a full-blown Mobile Library Web site (why not?)
Mobile Library: Built in LogiInfo
BUT – we quickly understood we needed some major infrastructure to fully realize this vision

Ø The Aleph application database is not built for reporting (not even close)

Ø It is even less suited to support linking to data in other databases

Ø The potential that hundreds of additional users (e.g. patrons) might be accessing Aleph data on a daily basis (though LogiInfo) had worrisome performance implications.

Ø The upshot – We needed a data warehouse for reporting and application purposes.

Ø ARC would not work because we wanted to integrate data from sources such as SFX, MetaLib and Verde
The Solution: LogiETL, LogiAdHoc and the Data Warehouse

Ø In mid-2009, we started discussions with LogiXML.

Ø In exchange for LogiETL and LogiAdHoc licenses, BC agreed to help LogiXML developers build a generalized data warehouse.

Ø Initially, we are focusing on a warehouse for the Aleph database.

Ø The warehouse can be extended to include data from SFX, MetaLib, or, indeed any other related data source.
The Aleph Data Warehouse Software and Architecture

Ø LogiETL: Used to extract, transform/normalize and load Aleph Data into the data warehouse
LogiETL Development Studio

This is the top-level element for any ETL job definition.
The Aleph Data Warehouse Software and Architecture

Ø Data Warehouse: PostgreSQL (Open Source Database)

Ø AdHoc Server: VMWare Virtual Linux Server (easy to bundle and distribute)

Ø LogiAdHoc: Allows non-techies to easily build reports as they require.

Ø No client installation. Interface is purely web-based

Ø Users can create charts, graphs and dashboards

Ø Reports can be scheduled and automatically delivered

DEMO
Thank You

• Questions?
• contact: kevin.kidd@bc.edu