

Home > 2006 > 09 > 11 > **Finding the Right Road UBC looks at a new Information System for Students**

Finding the Right Road - UBC looks at a new Information System for Students

By Ted Schellenberg



The new open source Student Services System promises to provide a lot more useful information to students and administrators.

About a year ago, top information technologists from around the world gathered in Orlando, Florida for the annual Educause conference. Included were representatives from a variety of professional levels and educational institutions, including The University of British Columbia.

It became clear during a key meeting at that conference that Student Systems were evolving to a new level, and that a 'turning point' was near. The broad adoption of open standards, the emergence of Open Source application software, and the ability to use distributed services connected by the Internet – all seemed to be leading to a new generation of application systems

in the field of student admissions, registration, and financial records.

It was felt at the Orlando conference that the time was probably right for the next generation of Student Systems. A subsequent feasibility study confirmed that "...there is broad-based support among university registrars, admissions officers, and technology specialists for such a system. In short, there is a growing acknowledgement that the time is right for colleges and universities to conceive and deliver a community source Student Services System (SSS)."

Wheeler Report Provides a Roadmap

Today, things are starting to heat up as SSS moves from the back to the front burner at many universities.

Brad Wheeler, the CIO of Indiana University (who recently spoke at our Town Hall event) released a report on "Open Source Student Services Systems" for the Andrew Mellon Foundation in July –

The report reflects an emerging vision of a community source SSS - articulated by UBC's Richard Spencer. The community source process is akin to open source in that the source code software comprising the system is openly shared with others; it is different, in that there is more of a directed project approach in the community source model.

Old Versions vs New

There are several characteristics of a community source SSS that distinguish it from the current generation of Student Information Systems (SIS):

- A primary focus on the needs and indeed, the point-of-view, of the students and faculty. (Many current systems are mainly concerned with back-office administrative functions)
- A rethinking of the traditional scope and functions of the system. (Today's systems often don't adequately handle non-traditional courses and programs, non-credit courses, or related areas such as athletics and housing)
- A flexible, modular architecture that will make it much easier for institutions to support new processes, and address changing business and legislative requirements.

"This is all achievable," insists UBC's CIO, Ted Dodds. "The technical platforms, infrastructure, and expertise needed to create such a system exist today. We believe that a 'next generation' Student Services System can be planned and delivered in a series of discrete phases over the next several years."

SOA Far, SOA Good

The key to making it all happen is something called *Service Oriented Architecture*, or SOA (http://en.wikipedia.org/wiki/Service-oriented_architecture). "By adhering to the tenets of SOA and its related methodologies," says Dodds, "we can build the new system using well-defined services - autonomous, reusable software programs that can be orchestrated to reflect the requirements of each campus...programs that are highly modular, can be designed to be reused across multiple business processes, and can be configured to accommodate changing or new business requirements."

In the meantime, those existing Student Information Systems are aging, and university administrators have to think about how they're going to keep those systems young and strong and reliable. They need to replace systems that use old technology and are increasingly difficult to support.

Do they continue buying and supporting complex Enterprise Resource Planning systems? These are the packaged vendor systems like PeopleSoft and Banner that may involve vendor lock-in, and upgrade schedules that don't always fit the plans, or meet the needs, of the educational institute.

Do they develop a large, in-house, one-institute Student System, an approach that comes with its own risks?

Or do they share development costs by building a community-sourced Student Services System?

The timing seems right for the latter approach, as we now have Service Oriented Architecture that facilitates the latter kind of development by allowing independent modular development by partners, using common data models and compatible middleware. Many of the institutions that participated in the feasibility study strongly favour an incremental upgrading or replacement of their existing system, using modular (and flexible) SSS components as opposed to the alternative solution – development and installation of a large new monolithic student system.

Learning from Past SOA Success

UBC's Ted Dodds agrees that the timing is right for this initiative: "The growing track record of *JA-SIG*, the *Sakai Project*, and *Kuali Financials* is giving colleges and universities the confidence that the community source software development model for higher education can also be used for a new Student Services System."

When will the open source SSS project be launched? Are we ready to develop a next generation Student Services System using the Community Source Model? We're close to making some major decisions regarding this initiative. We'll know more after those technology-oriented thinkers get together at the next Educause conference in Dallas this October.

Further reading: Educause Quarterly article on SSS
[<http://www.educause.edu/ir/library/pdf/eq/a002/eqm0022.pdf>]

Educause Website: <http://www.educause.edu>

This entry was posted on Monday, September 11th, 2006 at 10:31 am and is filed under e-Community.