EDUCAUSE Center for Applied Research

Research Bulletin

Volume 2007, Issue 19

September 11, 2007

Next-Generation Administrative Systems:

Philosophy, Principles, and Technology

Ted Dodds, The University of British Columbia

Richard Spencer, The University of British Columbia

4772 Walnut Street, Suite 206 ◆ Boulder, Colorado 80301 ◆ www.educause.edu/ecar/

Overview

It is time to think differently about administrative systems in colleges and universities. Higher education institutions have made, and continue to make, substantial investments in these systems, which have helped them run increasingly complex business processes, handle growth in key areas, and offer services to end users. However, today's systems are not always focused on the needs of end users. Students and faculty often feel the systems they use do little to simplify administrative tasks, and staff members outside core departments frequently feel systems have made their work more difficult and time-consuming.

Most current administrative systems, whether vendor-supplied or developed in-house, are inflexible, in part due to their monolithic design. Changes to support new business processes or new programs and activities can be complex and expensive. Although enterprise resource planning (ERP) systems have helped address the problem of separate systems with separate data stores, duplication of records, and processes that cannot be fully supported if they cross departmental boundaries, many institutions (including those with ERPs) continue to face these problems. Consequently, many institutions have shadow systems supporting similar processes in different business or academic areas.

In some cases, the implementation of a new system has led to a loss of functionality because applications that had been added to older systems, either through customization or in-house development, are missing from the new system. In a 2002 ECAR research bulletin, King, Kvavik, and Voloudakis (2002) point out that customization of ERPs, which can increase customer satisfaction, also increases the cost of installation and maintenance.

Vendor-supplied systems often require expensive and complex upgrades as a condition of continued support. Institutions are often disappointed when these upgrades—which are as likely to be driven by vendor strategies as by the business requirements of the customer—fail to deliver significant improvements to either business processes or support for end users. The increasing complexity of in-house systems has made it difficult for institutions to enhance them to support new processes and activities or to upgrade them to use current technology. Consolidations in the vendor marketplace and the increasing difficulty of supporting in-house developed systems have restricted the options available to institutions.

It is time to set ambitious goals for a new generation of administrative systems. Continuing exponential growth in the availability of computing processor cycles, memory, storage, and network bandwidth, together with the emergence and rapid growth of the World Wide Web and Web-services technologies, have made it possible to develop modular, flexible systems using a service oriented architecture (SOA). Community source development allows groups of institutions to share the cost and risk of developing new systems. Open source development, the adoption of open standards, and the publication of service interface descriptions can encourage the formation of communities, including both vendors and institutions, which will extend the support available for the new systems.

Institutions should expect more from their investments in administrative systems. They should expect systems that allow customization to meet local needs, make it easier to change business processes and support new activities, and support business processes that span departments and institutions. Most importantly, they should expect these systems to help students, faculty, and other end users complete administrative tasks, plan their activities, and achieve their goals.

Using the Kuali Student Service System (Kuali Student, for short) as an illustration, this research bulletin discusses a vision, and a set of functional and technical principles, for the next generation of administrative systems. Although the vision and principles are being developed for a new student system, they can serve as a framework for the development of an administrative system in any area of higher education. They are also a guide to what to look for in planning to select and implement a next-generation vendor-supplied administrative system.

Highlights of Next-Generation Administrative Systems

Kuali Student is a next-generation student system currently being developed through the community source process. The success of other community source initiatives, and the creation of organizations such as the Kuali and Sakai Foundations, show that institutions can form partnerships, agree on goals, make commitments, and share resources to create, implement, and sustain complex, open source administrative systems that meet the needs of colleges and universities. This is what makes community source development work.

Flexible student systems, combined with imaginative business policies and processes, can be a source of comparative advantage for an institution. However, institutions wanting to enhance their student systems using a “best of breed” approach find that individual applications seldom share a common architecture or consistent standards. Kuali Student will be an open source application, developed using an SOA. Publication of the code, service contracts, and service interface definitions will result in a community of institutions and firms that support Kuali Student and make it possible for institutions to use Kuali Student modules to extend the functionality of existing systems, without the need to replace an entire system or adopt products from a single vendor.

Kuali Student is an ambitious project that will face functional and technical challenges. The functional principles, design elements, and technical principles for Kuali Student are summarized below.

Functional Principles

The Kuali Student Functional Steering Committee has refined and enriched the original vision for Kuali Student through a set of functional principles. The five principles that follow are based on the work of the committee.

Support End Users

The system should make it easier for end users to do the things they need to do by anticipating their needs; helping them make choices, set goals, and track their progress; and reducing the time it takes them to complete administrative tasks. It should remind users about things they have to do, present them with options and suggestions that reflect applicable rules, and make it easier for them to decide what to do when they have choices.

Consider a new student registering for her first year of study at a large university. At most universities today, she would use a Web application that makes it easy to register in any course that has seats available and for which she has the prerequisites. She might be able to use an online degree audit system to see if a particular course will satisfy her degree requirements, but at many universities she would be responsible for figuring this out for herself, based on degree requirements printed in the university catalog. She will see hundreds of available courses and will find few sources of guidance or advice on which to choose.

Kuali Student will use information in the system to help this new student choose her courses. This information includes her high school courses and grades, her academic interests or plans (which she has entered in her portfolio), information about courses in which other students with similar background and interests have been successful, and the regulations that apply to her degree. Rather than simply enabling course registration, Kuali Student will support the student through the registration process, suggesting courses that match her interests, reminding her of applicable degree requirements, and making it more likely that her first semester will be successful.

Support a Wide Range of Users and Activities

The system should support the broadest possible range of users and activities across a wide range of institutions. The system should make it as simple as possible to support new activities and initiatives, rather than defining the users, activities, and types of institution that can be supported and then expecting institutions to define their users and activities to fit this framework, which requires expensive customization to support new types of user or activity.

In many colleges and universities, two separate systems are used to manage students, courses, and enrollment in credit and noncredit divisions. A third system might be used to manage the registration of staff in career development courses, and a fourth to offer online registration for participants in seminars and conferences. Kuali Student will be able to support any activity in which people engage in any form of learning, across all departments, whether academic or nonacademic, credit or noncredit.

Kuali Student will also support activities that are not supported by any of the student information systems currently in use on most campuses. A student who presents a short seminar in an undergraduate course will be able to add this to his or her learning record, for inclusion on a cocurricular transcript. Students helping to deliver noncredit, postsecondary-level courses for disadvantaged residents could have this recognized in

their learning record. Kuali Student will have the flexibility to handle all kinds of learners and learning activities, including new types of courses and programs introduced in the future.

Support a Wide Range of Business Processes

Many current systems reflect the paper-based, department-centric processes—with the redundant collection of information and resulting duplication of records—on which institutions depended for so many years. Processes that cross departmental, business, or system domains are often not well supported.

A new system should have open, documented interfaces so it can interoperate with other systems to support processes that cross departments, systems, and institutions. It should allow departments to retain full responsibility for their parts of a process and make it easier for them to work with other departments to provide users with cross-department support.

Because students' eligibility to compete on sports teams is usually dependent on both their academic record and their performance on the field, the process of ensuring that athletes remain eligible crosses the boundary between the athletics department and the registrar's office. By maintaining a single, high-level identity record for every person on campus and offering managed, secure interfaces to student records, Kuali Student will make it relatively straightforward to implement a business process that uses information from both areas, while respecting the authority and responsibilities of each.

Make It Easier to Change Business Processes

Institutions must be able to innovate and do things in new and different ways, but changes in business processes and activities can be difficult to support with existing systems. The new system should make it easier, faster, and less expensive to change existing processes and activities and introduce new ones.

At The University of British Columbia, a course approval process that involves multiple levels of review and often took eight months or more to complete was redesigned so that the entire process, from proposing a course for approval to making it available for online registration, could be completed in five days or less. The responsible decision-making bodies gave the new process their enthusiastic blessing, but several years later, the resources required to make changes to the student system to support the new process have still not been found. Kuali Student will greatly reduce the work required to change the rules and workflow that define this processes and others like it.

Enhance Human Interactions

An administrative system should reduce the time staff members spend on repetitive clerical and administrative tasks. It should also make services developed for end user self-service available to staff who provide in-person support to end users. This will allow staff members to provide higher-value support and services to students, faculty, and others who need them.

An online service that allows prospective students and applicants to match the admission requirements for various programs against their own record of courses and grades can also be used by staff working face-to-face with students. By enabling many students to explore options on their own, this service will reduce the number of inquiries made to staff. Because it applies the relevant rules accurately and consistently, its use will increase the number of staff qualified to provide in-person advice to students. Reducing the number of students who need in-person support and increasing the number of staff who can provide support can lead to dramatic improvements in service quality without the need for more staff.

Design Elements

The Kuali Functional and Technical Steering Committees have developed a list of seven elements that are critical to delivering the vision outlined above.

High-Level Entities

The high level of abstraction of a number of new entities is critical to making it easier to introduce new programs and approaches to learning. The entities include:

- *Person:* Systems often recognize individuals only in roles such as student, instructor, and so forth. People should be recognized as unique, enduring, individuals who will have various roles, associations, and group memberships over time. The high-level entity should be the person. Over time, each person can have one or more roles as student, instructor, staff member, research associate, and others.

- *Time:* Predefined blocks of time such as semesters, terms, and classes, with fixed start and end dates and/or times, do not fit all programs and activities and are often difficult to change to accommodate nonconforming activities. Time blocks should be defined using normal time units (year, month, day, hour, and minute) so they can start and end at any desired date and time.

- *Learning units:* Kuali Student will introduce learning units, a concept borrowed from inventory management. Learning Unit Numbers, or LUNs, are analogous to the stock keeping units, or SKUs, that make it possible for stores to manage huge, varied, rapidly changing inventories. LUNs can be assigned to any learning activity, including those that many existing systems are unable to handle. For example, a sequence of courses can be defined as a unique offering, given a LUN, and made available through the registration system. Contributions by students in courses can be identified with unique LUNs and added to the student's record. This would be difficult or impossible with existing student systems. The learning unit illustrates how high-level entities will allow systems to handle programs, processes, and activities that did not exist when the system was designed.

Concierge

The concept of a helpful agent is critical to delivering a user-centric system. Kuali Student will include a “concierge” application that supports end users by anticipating their needs; alerting them to necessary tasks; presenting information about available choices and courses of action; helping them make choices, set goals, and track their progress; and reducing the time it takes to complete necessary tasks. The concierge makes use of information about end users (some of which they store in a portfolio or profile, and some of which is in one or more systems); knowledge of institutional rules, regulations, and learning opportunities; and information about the experiences of other learners in similar or related situations. Using rules and workflow, the concierge will identify required or recommended actions and present relevant alternatives with information about the probability of success, based on the experience of others with similar achievements and goals.

Rules Engines

Many administrative processes in higher education are labor-intensive and do not scale well to serve increasing numbers of users or increased transaction volumes. Many institutions find that growth in areas such as student recruitment, or the number of research grants received, can require substantial increases in staff. If the required resources are not available, service levels usually fall, and users become dissatisfied and frustrated. Scalability, or the ability to support additional users and processes at low incremental cost, requires automated, self-service processes that serve end users directly, applying rules accurately, consistently, and with minimum human intervention. Rules engines should be used to ensure that the system is highly scalable and that changes in rules can be implemented in a cost-effective and timely manner.

Workflow

As noted by King, Kvavik, and Voloudakis (2002), customizing an ERP results in an implementation taking longer and costing more. Core business and academic processes evolve to reflect the role of the institution, its learners and programs, and its academic mission. In some cases it is appropriate to change these processes to match those supported by an ERP, but in other cases it is not. Workflow and rules engines will make it easier for institutions to implement processes in ways that work best for them—to implement their practices, not someone else’s “best practices.”

Modular System

A standards-based, loosely coupled modular design will make it possible to add other applications that either enhance included functionality or add functionality that is not included. For example, a portal will provide a user interface that can be configured to suit a wide variety of users and can present information from other systems. An SOA will make the core functionality on which higher level processes depend available to multiple applications and to other systems. This will ensure that processes that cross traditional

departments (and systems) can be supported. A modular design, together with an open source licensing model and the publication of service interface information, will allow both open source and commercial modules to be used with the system.

Access to Information

Users should have appropriate access to the information in administrative systems. Many existing systems make it difficult, and in some cases impossible, for all users to have timely access to the information they need, in a form that is useful to them. The modular, service-oriented design of a next-generation system, together with the use of an identity management system to delegate the authority to manage users and their roles, memberships, and privileges, will make it possible to give all users appropriate, timely access to information in a form that meets their needs.

Internationalization

To increase the potential for institutions and vendors in a number of countries to use the system and participate in building, supporting, and adding functionality to it, and to meet the needs of institutions that operate in more than one country, it is important that modules can be easily adapted for use in countries that use languages other than English, use non-U.S. currencies, and have different educational systems. A modular, service-oriented design, with the use of workflow and rules engines, will help make this possible.

Technical Principles

A number of technical principles have been developed by the Kuali Technical Steering Committee to guide the technical development of the Kuali Student system.

Service Oriented Architecture

- *Service oriented architecture:* SOA is based on the identification of “services,” reusable components that are defined by service contracts (what they do) and service interface definitions (how they communicate). SOA is fundamentally important to the new system.
- *Web services:* Web services offer simplicity, universality, and platform neutrality and are the method by which services communicate.
- *Standards-based:* The system is based on open standards, and open-standards compliance is a key criterion for selecting components used in the system.
- *Separate governance for service contracts:* Service contracts are the business assets of an SOA system. They are governed separately from the underlying technical assets, by a broadly representative body that will ensure the system can interoperate with other systems and support activities in a wide range of institutions and countries.

Component Abstraction

SOA provides an effective means of abstracting layers, which is achieved through application of the following principles:

- *Abstraction of business processes and business rules:* Rules engines are used to abstract business rules, and workflow is used to abstract business process logic.
- *Abstraction of the presentation layer:* Use of a standards-based portal as the user interface allows services to interact with users through portlets or channels, without having to provide their own separate user interfaces.
- *Abstraction of the data layer:* The data model is derived from simple abstractions representing concepts such as time, people, and, in the case of Kuali Student, learning units and related entities. Templates are used to implement these abstractions as domain objects. Services will provide data to other applications.

Leveraging Open Source

Kuali Student will be built using open source products and tools (which does not preclude the use of commercial software for individual implementations). This, together with the modular design, will lead to the growth of a community of developers and vendors that will add functionality to the system and broaden the range of institutions and countries in which it can be used.

- *Building an open source system:* The system will be built on an open source software stack compatible with the Educational Community License (ECL) and the Kuali Foundations IP management policies. Institutions can replace open source components with standards-compliant vendor-supplied components.
- *Building the reference infrastructure from open source products:* Existing open source components will provide the infrastructure services, such as workflow or rules engines, and an enterprise service bus.

What It Means to Higher Education

The next generation of administrative systems in higher education

- will use an SOA architecture and Web services,
- can be built on an open source platform using open source products and development tools,
- can be developed as community source projects within higher education, and
- will run on more powerful and scalable hardware and networks.

Focus on the End User

The new administrative systems will focus on meeting the needs of end users, particularly students and faculty, two groups that are not well supported by many current administrative systems. The use of rules engines and workflow will make it practical to implement more processes and deliver more services online, directly to end users. This will reduce or eliminate the need for human overview, rules checking, and approvals and will lead to fewer shadow systems. Abstraction of the data layer will allow systems to share data, eliminating redundant data entry and storage. Processes and services will be scalable, delivering the same consistently high standards of service to all users, with low incremental increases in cost as the number of users increases. Staff will have more time to support people and better tools for providing that support.

Managing Change

The technical challenges of designing, building, and implementing next-generation systems can be overcome. However, managing the change required to take full advantage of these systems will be equally challenging. In spite of being committed to the discovery of knowledge and the education of people who can change the world, higher education institutions are often not leaders in using technology to improve their processes and their service to end users. Institutions should not consider implementing next-generation administrative systems unless they are also committed to making radical changes in their processes and services.

Flexible Support for Processes and Activities

The new service-oriented systems will make it easier to support processes that have either evolved to meet specific needs or have been redesigned to improve service and lower process time and costs. Reusable services with open, published interfaces, together with component and data abstraction, will help break down the silos that still exist between systems and allow support for processes that span departments and institutions.

The use of new high-level entities will make it easier to support new kinds of teaching and learning and to implement systems that support the full range of teaching, learning, and research activities in an institution. The modular design will make it easier to add functionality that may be missing from current ERPs and to incorporate both open source and vendor-supplied components.

Open and Community Source Development

Systems can now be implemented using SOA and Web services, on an open source platform, using open source tools. This allows institutions to consider alternatives to both vendor-supplied and in-house developed systems. Collaborative, open source system development, using the community source model, with the prospect of support from both open source communities and vendors, is a realistic option. This option should not be chosen to reduce either system costs or the time for development and implementation, and it is not free of risks. However, it can offer higher education institutions more control

over future costs and some protection against changes in the vendor marketplace over which they have no control.

Key Questions to Ask

- In what areas are your students, faculty, and staff satisfied with the way your administrative systems help them carry out their work or achieve their objectives, and in what areas would they like these systems to do more?
- Which processes or activities at your institution would benefit from greater support from your administrative systems or ERP?
- What functionality do users want to see added to your current administrative systems or ERP?
- How successful are you in getting past the hype surrounding service orientation and investigating its methodologies and technologies?
- What capacity and motivation exists at your institution to contribute to an open source or community source systems development project?
- How successful is your institution in managing change and introducing new business and academic processes?
- Which administrative systems, if any, is your institution considering upgrading or replacing?

Where to Learn More

- ”Delivering the Kuali Student Functional Vision,” <http://www.kuali.org/assets/pdf/KS-DeliveringtheFunctionalVision.pdf>.
- “Kuali Student Technical Architecture Principles,” <http://www.kuali.org/assets/pdf/KS-GuidingPrinciplesforTechnicalArchitecture.pdf>.
- Kuali Foundation, <http://kuali.org/>.
- Sun Education Commons. (2006). *Community source student services system*. Retrieved May 10, 2007, from http://educationcommons.org/projects/display/CSSSS/Home
- Wheeler, B. (2004, July/August). Open source 2007: How did *this* happen? *EDUCAUSE Review*, *39*(4), 12–27. Retrieved May 10, 2007, from http://www.educause.edu/ir/library/pdf/erm0440.pdf
- Wheeler, B. (2007, January/February). Open source 2010: Reflections on 2007. *EDUCAUSE Review*, *42*(1), 48–67. Retrieved May 10, 2007, from http://www.educause.edu/ir/library/pdf/erm0712.pdf

Reference

- King, P., Kvavik, R. B., & Voloudakis, J. (2002, November 12). *Enterprise resource planning systems in higher education* (Research Bulletin, Issue 22). Boulder, CO: EDUCAUSE Center for Applied Research. Available from http://www.educause.edu/ecar/

About the Authors

Ted Dodds (Ted.Dodds@ubc.ca) is Chief Information Officer and Richard Spencer (Richard.Spencer@ubc.ca) is Senior IT Strategist at The University of British Columbia.

Copyright 2007 EDUCAUSE and Ted Dodds and Richard Spencer. All rights reserved. This ECAR research bulletin is proprietary and intended for use only by subscribers. Reproduction, or distribution of ECAR research bulletins to those not formally affiliated with the subscribing organization, is strictly prohibited unless prior permission is granted by EDUCAUSE and the authors.