NSU NOVA SOUTHEASTERN UNIVERSITY Abraham S. Fischler School of Education

Innovate: Journal of Online Education

Volume 3
Issue 1 *October/November 2006*

Article 4

11-1-2006

Vision 2010: The Future of Higher Education Business and Learning Applications

Patrick Carey

Bernard Gleason

Vision 2010: The Future of Higher Education Business and Learning Applications

Vision 2010:
The Future of Higher Education Business and Learning Applications

by Patrick Carey and Bernard Gleason

The global software industry is in the midst of a major evolutionary shift—one based on open computing—and this trend, like many transformative trends in technology, is being led by the instructional technology (IT) staffs and academic computing faculty of the higher education industry. The elements of this open computing approach are open source, open standards, open architecture, and open communities. Such an approach does not only involve the use of open source software code; the application code (whether proprietary or open source) must also be developed to adhere to open data and content standards, designed to operate within an open architecture, and, in most cases, designed and developed based on requirements determined by a larger open community. Together, these elements hold the promise of the next generation of business and learning applications for higher education. If history does indeed repeat itself, we believe that this trend will have the same effect on most, if not all, other industries.

Such a trend may be seen not only in the emergence of various open software applications over the past few years but also in the adjustments that proprietary developers have made in their own business models. Future-focused commercial vendors are embracing open standards and open architecture, even when they are not open-sourcing their application code. In doing so, they recognize that open standards and open architecture simplify the processes of developing or acquiring new applications—and, in turn, of integrating and customizing client systems to match evolving business practices. They see that basing applications on open standards can accelerate the deployment process, ease adoption by more users, improve operational productivity, and reduce costs. They are also responding to significant patterns in user behavior that have already begun to change the playing field for software developers. For example, in analyzing the results of its February 2006 survey of U.S. higher education chief information officers (CIOs), the Alliance for Higher Education Competitiveness (A-HEC) found that "two-thirds of CIOs have considered or are actively considering open source with a leading edge of about 25% of all institutions actively engaged in implementing higher education specific open source applications of some type"; in response to these trends, the A-HEC concluded that the "value proposition for open source applications can be summarized as a combination of cost, control, and possibility for innovation" (Abel 2006, 5). In the commercial world, InformationWeek's November 2004 survey of more than 400 IT professionals found that although only 2% across all industries were using open source software primarily, two thirds were actively using open source, and three in five reported an environment of both open source and commercial software (D'Antoni 2004).

Such findings are consistent with our own observations at IBM. As a major and long-time participant in the open movement and as one of the world's largest software companies, we expect the future technological environment to be made up of a combination of open and closed source code, all of it adhering to open standards, thereby reducing the costs of integrating and operating the software by client institutions. At present, however, many institutions that have successfully implemented enterprise resource planning (ERP) systems and course management systems (CMS) still face the challenges of streamlining their operating procedures, reducing the financial and operational burdens of such systems, and ensuring that such systems can be made flexible enough to adapt to new business needs and technological innovations. Colleges, universities, and commercial software providers alike will need to formulate a clear vision of the factors that for the next five years will shape business and learning applications as well as the architecture in which the applications operate. To this end, we outline and discuss several key developments on the horizon of open computing that university planners should anticipate as they envision the future of their technological infrastructure.

The Future of the Open Movement in Education

In the next five years, what business models will higher education institutions adopt? What direction will application software development take?

Many predictable outcomes and trends will emerge. For colleges and universities to stay competitive, offer exceptional learning experiences, attract professors and students, promote research, and compete globally, they should consider adopting an open approach in designing or obtaining both business and learning application software. An open approach has at its core a focus on reducing the costs of the integration and operation of campus business and learning systems, empowering the institution to redirect scarce resources into improving the academic and research environment. The evolution of an open-based movement in higher education will unlock the innovative talent, forces, and flexibility that institutions desperately desire. The following trends will be particularly important for university planners to recognize as they formulate strategies for the future.

Modular not Monolithic

Applications developed an open, standards-based architecture are often assembled from components that are either freely available or less costly than comparable proprietary packages. Instead of relying on monolithic design as in the past, developers using an open approach rely on interorganizational or intraorganizational networks on the Internet to find functional modules loosely coupled as sets of services within a service-oriented architecture (SOA). The architecture is deemed service-oriented because applications and functions within applications can be joined and choreographed to create composite applications and business processes; this approach allows components to be selected and joined to create more precisely the composite applications and processes needed by an institution.

Reusable component software is available either at a price or for nothing from a variety of sources such as commercial independent software vendors (ISVs), online open source repositories, or subscription from a hosted service. For example, higher education groups such as the Java Architectures Special Interest Group (JA-SIG) sponsor open source projects and host repositories of applications and documentation that are freely available for downloading by all interested colleges and universities. In turn, IBM developed a framework for Java development called Eclipse and, rather than compete with other Java tool providers and require each company to create its own framework, contributed the code as open source and funded the start-up of the independent Eclipse Foundation. Today all major Java toolmakers are members of the foundation; each company is saving money and is able to rely on access to a consistent environment for open source development.

Through such open approaches, ISVs still create, assemble, sell, and support software solutions, but vendor differentiation shifts to innovation in applications and quality of support, particularly in integration services. IBM and other major vendors of enterprise resource planning solutions are moving in this direction, and next-generation course management systems such as the Sakai Project are already emerging to meet similar needs. However, continued progress is likely to be constrained by the requirements of the installed base and by the commercial vendors' desire to protect their investment in existing customer relationships and license maintenance streams. Consequently, university planners will need to exercise discretion in the systems they adopt; if a compatible open source solution cannot be found to address a particular need, they should still select proprietary software that utilizes open standards within an open architecture and thereby allows for modular development.

Community Open Source Model

Today, groups of colleges and universities are pooling application development and financial resources to create common sets of applications. This approach is called community open source. The community open source model is built on the basic principles of open source software development—that is, code is freely

distributed and users are free to modify it. The main distinctions of the community source model are the methods for funding, managing development, and guaranteeing sustainability.

The community source model is favored by the Andrew W. Mellon Foundation and is exemplified by the Sakai Project (course management system) and the Kuali Project (financial system). Both projects leverage code already developed by universities to jumpstart the creation of a new breed of open source enterprise applications. Their model functions in the following manner:

- Foundations provide start-up funding to be matched by participating institutions.
- A core group of institutions supplies oversight, central development resources, and funding.
- A larger group of institutions makes smaller annual financial contributions and participates in defining requirements, testing, and evaluation.
- Core developers and project leaders are compensated for their work.
- Commercial partners contribute technical resources and offer customer support services.

Institutions seeking to benefit from the possibilities of open source computing should give special attention to projects such as these since they have established a strong foundation for sustained innovation and lasting value. These value networks, sometimes unforeseen and unpredictable, will continue to sprout up within the higher education community as individuals and units with common interests and requirements come together. Consortia and associations of institutions (e.g., all colleges in a conference) that already have common-value structures are likely to be the early winners in sharing resources and exercising group influence and buying power.

Meanwhile, for all community open source projects that adhere to the concepts of an open approach—open standards, open source, and open architecture—the next logical step is the creation of a coordinating body, similar in concept to the Apache Software Foundation. Ira Fuch, Vice President for Research in Information Technology at the Andrew W. Mellon Foundation, similarly advocates the formation of an open source congress to coordinate the interproject collaboration, communication, and certification of multiple related projects within the higher education ecosystem (cf. Currie 2005). Such a coordinating body helps ensure the placement of a system to maintain and enhance the code over time, a major concern to any campus administrator considering an investment in a critical business system. As further advances are made to establish sustained coordination and development of open source projects, institutional planners will be able to make informed decisions about the potential adoption of such technologies.

Open Integration Framework for Education

One of the biggest challenges facing application developers is likely to be integration and interoperability—getting applications to talk to each other. Institutions can purchase or internally develop an SOA, but colleges and universities are not going to be able to quickly re-engineer existing applications and processes to be SOA-compliant and compatible with one another. The answer to this challenge lies in the creation of a new breed of open applications that will provide a flexible and scalable technical framework for supporting open standards and open application interfaces.

At the heart of an SOA is something called an enterprise services bus (ESB), a kind of middleware—software that operates between an operating system and an application—that utilizes standard data definitions and messaging formats to transfer data securely between different programs. An ESB supports the running of existing applications while adding new functionality and adapting to a new architecture based on Web services, and it is likely to be the first step in the shift to the full adoption of SOA and a modular design for connecting applications. The ESB will allow clients to follow an evolutionary approach in moving from a dependence on a single software provider or a proprietary technology platform to an open computing approach.

We strongly believe that an open integration framework for higher education will emerge that is based on an open source ESB from IBM and perhaps others. The open integration framework will employ commodity middleware, will support an extended set of trusted services, and will be customized to support the special data and content standards for higher education business and learning. A logical first instance will be in the area of student services—a Student Services Bus. The Student Services Bus, in combination with an existing student information system and learning management system, will provide a platform for integrating new services with core functions; in doing so, it will free institutions to think and act beyond the boundaries of traditional student learning systems and beyond the notion that all functionality must be delivered by one provider (cf. Carey and Gleason 2005). As university planners seek ways to improve their technological infrastructure, the flexibility afforded by advances in ESB technology will open new possibilities for easing the transition to an open integration framework and enhancing the services they provide their students.

Superior Vertical Solutions

Until now, the commercial vendors of enterprise planning and learning software have promoted a common code base across all industries as a way to minimize their code development and maintenance costs. This rigid one-size-fits-all approach often restricts functionality, causing unnecessarily high costs and forcing institutions to adapt their business processes to the vendor's solution, instead of the other way around. As a result, educational institutions are often forced to change their existing business processes significantly or undertake expensive software customization projects.

However, in the new, open, and component-driven SOA, developers can begin to create or obtain superior solutions as customizable modules, which are designed as services to be plugged into the open, standards-based application architecture. The modular design makes it easier for developers to innovate and to create superior components without worrying about the total application system. For example, if one institution creates a Web-based course registration module that is superior in functionality—a module perhaps equivalent to Amazon in completeness and user-friendliness—another institution could adopt it easily by swapping out just the course registration module without having to replace the rest of the institution's existing student information system. If another institution creates an online testing module for a learning management system that requires minimal user knowledge of specialized design code, another institution could adopt it within its own learning management system without the need for replacing or extensively modifying the system itself. In such cases open architecture and open standards provide the foundation for greater flexibility as designers can incorporate different components from one system to the next.

Software-as-a-Service

Campus constituents—students, faculty and staff—are increasingly dependent on information and learning services provided by the colleges and universities. Their experiences with information and communications technology in other aspects of life have led them to expect reliability and 24/7 availability in campus systems. These requirements are pushing higher education institutions to provide additional support, which entails an increase in associated costs with no new supporting revenue. In some cases institutions have tried to reduce costs by deploying no-frills versions of ERP and CMS applications that do not require the same level of support. Yet by settling for lesser capabilities, such institutions may be compelled to sacrifice quality in order to remain relatively viable in the highly competitive area of student services.

In light of these challenges, many colleges and universities will need to rely on third-party providers in order to meet student demands more economically and on a level that cannot be duplicated by an individual institution. As they do so, an open architecture that utilizes open and universal security standards will serve as a crucial foundation to facilitate such partnerships.

1.) *Applications*—As colleges and universities look for new ways of controlling costs, reducing the burden on their technical staff, and simultaneously improving constituent services, we can expect resurgence in hosted

ERP and CMS applications. More importantly, the existence of an open architecture will facilitate the outsourcing of portions of the application base, and institutions will easily be able to swap out third-party service providers if they are not satisfied with the quality or price of the service. Likewise, such an open architecture would allow administrators and instructors to select from a variety of competing third-party providers in the design, customization, or support of specific CMS applications.

2.) *Integration as a service*—A major concern in adopting a service-oriented architecture is the effort required to maintain, manage, and monitor many application connections. Rather than worrying about services with a multitude of interfaces, higher education institutions may begin to consider outsourcing the integration layer. All connections to business processes from the institution would conform to a set of defined higher education industry standards for data, messaging, privacy, and security and would connect to an integration provider. Integration providers would, in turn, manage the service-level agreements and transactions with partner networks.

3.) *Outsourced processing*—Generally, every college and university performs the same administrative functions and has similar processes to support those functions. For example, every undergraduate admission process collects the same types of information, electronically and manually. Instead of collecting and scanning forms and translating data files in the back office, this work can be outsourced easily and cheaply to an agency that performs the same tasks for multiple institutions. In such cases, of course, institutions would ensure student privacy through public agreements with the service provider, and with such agreements in place, office professionals would find much more time for personally serving their constituents.

Conclusion

An open computing approach will generate practical innovation in the business and learning areas of higher education institutions along with the cost savings that colleges and universities are desperately seeking. Colleges and universities that aspire to be regarded as one of the best places to work or be educated must look beyond today's applications and operating models; these higher education institutions must be committed to the continuous evaluation and adoption of new technology innovations and to the pursuit of opportunities for collaboration in what has been historically a very collaborative environment. Developing a long-term strategy for administrative and learning applications is one of the most important financial decisions campus executives can make as it will have a significant impact on all members of the community for decades to come.

References

Abel, R. 2006. Best practices in open source in higher education study: The state of open source software. Lake Mary, FL: The Alliance for Higher Education Competitiveness, Inc. http://www.a-hec.org/media/files/A-HEC%20open%20source%20hed%20030106.pdf (accessed September 17, 2006).

Carey, P., and B. W. Gleason. 2005. Student services system—Next generation. White Plains, New York: IBM Corporation.

http://www-03.ibm.com/industries/education/doc/content/bin/IBM_BCS_White_Paper_Student_Services_System_FIN (accessed September 17, 2006).

Currie, C. 2005. The open source congress. *EDUCAUSE Review* (July/August): 80-81.

http://www.educause.edu/ir/library/pdf/erm05414.pdf#search=%22fuchs%20%2Bopen%20source%20organization%

D'Antoni, H. 2004. Open source software use joins the mix. *InformationWeek*, November 1. http://www.informationweek.com/story/showArticle.jhtml?articleID=51201599&tid=5979 (accessed September 17, 2006).