

Challenges and Opportunities of Open Source in Higher Education

Ira H. Fuchs

The combination of expanding demand for IT services on college and university campuses, coupled with flat or declining budgets at many institutions, is arguably the greatest systematic challenge facing higher education IT during the next several years. It puts pressure on IT organizations to juggle “keeping the lights on” with the need to provide innovative technology platforms that support new levels of collaboration and promote strategic agility for institutions as well as their faculty and students.

The pressures will only be exacerbated, for many institutions, by the arrival of federally funded, next-generation “cyberinfrastructure” that will require installation, adaptation, training, local administration, and support. If institutions do not have in place an IT infrastructure that can support these new arrivals efficiently, the cumulative burden on IT staff and budgets could be formidable, even unsustainable. Moreover, even if these cyberinfrastructure platforms are integrated, institutions face significant challenges in diffusing their islands of expertise broadly to the campus community.

One interesting potential strategy for dealing with these problems is the model of “community-source” software (CSS) production. Another is the move to a services-oriented architecture (SOA). Both strategies show significant promise, but it remains an open question as to how much of that promise can actually be achieved. The next few years will be crucial for both.

Community-Source Software (CSS)

During the last eight years, the Andrew W. Mellon Foundation’s Research in Information Technology program (Mellon/RIT) has made significant investments in CSS projects to build open source IT infrastructure, particularly learning management (<http://www.sakaiproject.org>) and enterprise resource planning (<http://www.kuali.org>) systems.

These efforts are continuing: The Kuali project released the first version of its Kuali Financial System (KFS) this year; KFS has just recently gone into production at its first nonfounder institution; and the Kuali community is organizing to deliver a student information system as well. Moreover, community source initiatives have now been initiated in the academic library IT community (<http://oleproject.org>) and for the support, via shared technology services, of scholarship in the arts and humanities (<http://www.projectbamboo.org>). The premise behind these projects is that community-source approaches result in better quality, better fitted, more sustainable, and mission-critical software for higher education. Mellon/RIT and others have made the case for CSS as an alternative to proprietary commercial or home-built software in many places during the last several years. I will not repeat it here, but interested readers are invited to explore further.¹

In a community-source project, a consortium of colleges and universities comes together to build software needed by each of them, contributing resources to a virtual organization that serves as a software development shop. As both designers and customers, they build software tailored to their own needs; the involvement of multiple, diverse institutions ensures that the product is usable by many other institutions as well, and Mellon provides support to offset the costs of collaboration and generalization. After the product is first delivered, control and governance over the project are released to the higher education community on an open source, participatory basis: Institutions contribute human and capital resources to maintain and enhance the software and govern the intellectual property of the project via participatory mechanisms. The open source nature of the project is critical: It protects institutions against vendor lock-in, allows them to share the costs of maintenance and enhancement widely, and permits widespread innovation at minimal cost.

Because CSS is designed and built by and for higher education, the “adaptation gap” between community-source software and a particular institution should be much smaller than that between commercial software (which is typically designed for the vendor’s much larger, for-profit markets) and higher education. The result should be software that is cheaper to install and customize and that, because it is open source, is also far cheaper over time to maintain and enhance. Those savings may occur immediately, but it is more likely that they will only occur over time, as the software matures. Potentially even more important, the resulting software should be better fitted to the needs of higher education than its commercial competitors, both because it is newer (a transient advantage) and because it is built for and by its own customers (a durable advantage).

However, community-source software is still new, so the question of whether it is truly cheaper and/or better fitted to the needs of higher education institutions is still open. The next several years will tell if the promise can be realized—but it would be to the advantage of higher education to move up this timetable as much as possible. If CSS delivers on its promise of substantial savings and/or improved outcomes, then the sooner institutions are made aware of that fact, the more resources higher education as a whole can save; if it does not deliver those benefits, then the sooner the participating institutions know, the sooner they can either correct the problem or seek an alternative, better strategy. In either case, large sums of money can be saved, across all of higher education, by quicker, better answers to the core questions surrounding CSS. A quicker answer would require a concerted effort by an objective party to bring together the higher education community to conduct a formative and summative assessment of one or more CSS projects.

Next, assuming that CSS proves to be advantageous for at least some purposes and institutions, there is a need for some coordinating entity to assist them with the next stages of growth. CSS projects require a variety of logistical assistance, from conference planning to legal consultation on matters of tax status and intellectual property licensing. Many of these needs were detailed in the Mellon-funded EDUCORE/OOSS study (http://www.ithaka.org/strategic-services/oss/OOSS_Report_FINAL.pdf). If CSS realizes its promise, the number of CSS projects is likely to grow to the point where the proliferation of CSS governance organizations could itself become burdensome to higher education. Collecting these entities under one umbrella would make possible substantial efficiencies in service delivery; doing so proactively would further increase savings and provide other potential synergies as well.

Services-Oriented Architecture (SOA)

Many higher education institutions are excited about the promise of services-oriented architectures for increasing flexibility and cutting costs. The opportunities are real, but there are significant challenges as well. One significant opportunity is the chance to reduce the increasing burden resulting from the steadily growing number of “siloes” applications that a campus must support. Reconfiguring these applications into services running in a common environment can drastically reduce the amount of redundant software code that institutions must support, cutting costs and improving the ease of collaboration.

Another opportunity is the chance to accommodate newly arriving cyberinfrastructure in a way that is affordable and sustainable. Without SOA, each cyberinfrastructure project would need to be treated as its own silo and adapted to the campus (and to all other cyberinfrastructure projects) individually; with SOA, one can hope to integrate each new project more painlessly into the common enterprise architecture. The cost difference between the two approaches may make the difference between unsustainable and sustainable cyberinfrastructure for many higher education institutions.

Yet another opportunity is the promise SOA holds for reducing boundaries between higher education campuses. In an increasingly global higher education community, technology is too often a barrier to collaboration and the exchange of ideas. SOA holds some promise of lowering those campus boundaries.

However, even SOA enthusiasts only claim that SOA makes possible the benefits they enumerate; few would claim that SOA makes such benefits inevitable. Pursued with insufficient forethought, or even just campus by campus, SOA is less likely to deliver on its promised benefits. SOA is as much an organizational as a technological innovation and, as such, careful design, training, governance, and support are all crucial to its success. In particular, campuses that do not understand their own business models are at a significant disadvantage when implementing SOA. Without careful forethought, applying SOA locally to a single application or a single campus may simply copy the problems of the current IT infrastructure into the SOA design. Even with such forethought, SOA designed and built campus by campus is far more likely to replicate today's campus boundaries in tomorrow's technology than is SOA designed and built by institutions working together. If SOA is to provide the multi-institutional collaborative support that higher education users are demanding, it will need to be built by higher education acting as a whole. Even if individual campus efforts could be coordinated retroactively to achieve nearly the same collaborative result—which is by no means certain—that path would require resources and time that higher education can ill afford.

These opportunities and challenges suggest a role for a coordinating and educational entity around the issue of SOA in higher education. Such an entity could make it a priority to help higher education institutions equip themselves with the knowledge and skills required to make a successful transition to SOA. A combination of training and consulting services could help IT leaders with process and project management skills, as well as providing materials to educate key campus stakeholders in the costs, benefits, and skills required to make the SOA transition.

At a deeper level of engagement, the entity could assist in coordinating SOA design exercises among many higher education institutions. These exercises could be modeled on the design workshops developed by the CSS community (<http://educationcommons.org/projects/display/CSSSS/Home>) and could bring together diverse institutions to analyze their respective business processes and develop specifications for SOA-based higher education enterprise IT environments that could meet the needs of institutions of every size and type. Moreover, the entity could provide coordination and support for consortia and virtual organizations working to build the software according to the workshop designs. This role is also anticipated in the EDUCORE/OOSS study; it suggests that a single entity might be able to fulfill the mentoring and steering role for both CSS and SOA, going forward.

Diffusing Cyberinfrastructure

The challenge posed to campuses by incoming cyberinfrastructure is a variant of an old campus problem: how does an institution take the technological expertise concentrated in relatively small islands on campus—historically, the computer science department and perhaps one or two others—and diffuse it broadly, to provide equal support to, say, faculty and students in the arts and humanities? As one wag put it, getting technology delivered to the arts and humanities in higher education today is a little like trying to get a pizza delivered to a house in the suburbs ... in 1927.

Because of Mellon's long-standing commitment to the arts and humanities, we consider this a particularly important question. Our best current answer has two parts: Infrastructure is essential, and it must be designed for the recipients, not the donors, of the expertise. One cannot deliver technology *ad hoc*, to unsophisticated users, and expect much sustained benefit. Even if one deploys the training and support resources required to make that piece of technology a success, users wishing to grow beyond that project will still have nowhere to go. For sustainability, such projects must be embedded in an infrastructure that reduces the cost of generalizing expertise gleaned in one project for use in other projects—and that reduces the costs of mounting those other projects so that each one need not require extraordinary effort.

All this will still not be enough if the infrastructure provided does not accommodate the actual users. For example, most NSF-funded cyberinfrastructure projects are built by and for scientists. They assume a level of technical competence and programming sophistication that does not exist in arts and humanities disciplines; in some cases, even

the metaphors that underlie the user interface are scientific and, as such, foreign to humanists. Even if one could strip out the scientific content of such a project and replace it with humanistic content, those platforms could never serve as an adequate infrastructure for humanists today. Consequently, Mellon/RIT is supporting initiatives to build infrastructure by and for the arts and humanities. One such project is SEASR, the Software Environment for the Advancement of Scholarly Research (<http://seasr.org>), which is currently in beta release. SEASR provides an analytics platform that is as capable as many of the cyberinfrastructure platforms in the physical and biological sciences but differs in two key respects: (1) It is optimized for the analysis of unstructured data (text and rich media), rather than the structured data of scientific projects, and (2) it is intended to have a humanist-friendly user interface—one that uses language, images, and metaphors that are familiar and intuitive to people with no mathematical or programming backgrounds.

SEASR is one step in a planned series of initiatives culminating in the development of a comprehensive technology platform for the arts and humanities. This platform will support the management of scholarly content and collaboration throughout the scholarly life cycle—from creation, to collaboration, to review, to publishing. Experience suggests that such a platform, if built for the least technologically sophisticated disciplines, will be just as useful and easily adopted/adapted by more sophisticated disciplines—where the reverse is far from true. That is a lesson that any higher education institution wishing to ensure parity of technology support among all its faculty would do well to remember.

CSS and SOA Together

One can use CSS models to build SOA projects; in fact, several initiatives are under way now to develop the necessary infrastructure to support SOA-based CSS projects. Mellon/RIT has funded the Kuali Student project (<http://student.kuali.org/>) as a “pure” SOA-based, next-generation student information system. As part of its work, Kuali Student will deliver the essential infrastructure required for an enterprise-quality, higher education-specific SOA environment. Also, the Kuali Rice initiative (<http://rice.kuali.org/>) is extracting essential shared technologies (including messaging, event notification, workflow, and a service bus) from the existing Kuali projects so that they can be used even by institutions that have not installed the Kuali applications; the Kuali Enterprise Workflow tool, in particular, has already achieved independent adoption in several other administrative capacities.

The SEASR project, already mentioned, is using SOA to support humanities computing projects such as MONK (text mining; <http://www.monkproject.org/>) and NEMA (music information retrieval; <http://nema.lis.uiuc.edu/>). Mellon/RIT also funds a user interface layer (<http://www.fluidproject.org>) and supports a data management infrastructure (for example, <http://www.fedora.info>, <http://wiki.fluidproject.org/display/collectionspace/>, <http://oleproject.org>). Other key pieces of SOA-based, CSS-built infrastructure, such as event and resource calendaring software, are planned. Each project uses open standards-based interfaces to permit it to be used integrally with the others—or with whatever mix of commercial and open source alternatives a particular institution might wish to employ.

One key objective of the projects that utilize a service-oriented architecture is to allow institutions maximal freedom to tailor an enterprise infrastructure that both serves their distinctive institutional needs and supports easy, standards-based collaboration within and across institutional borders. A second major objective is to provide an environment in which new demands on IT can be accommodated as they arise, in the most affordable, efficient way possible.

Conclusion: CSS and Coordination

Building a multiproject CSS environment introduces organizational and coordination challenges that are greater than those associated with any single project. The various projects must remain coordinated over time because they must continue to work together effectively. Their licensing and other requirements must be congruent, so that institutions are not caught between warring policies. Perhaps most important, their administrative overhead must be managed in such a way that it does not become burdensome to institutions to support the entire environment. Given the significant overlap among these initiatives, the current practice of creating a new organizational entity to oversee each project is a form of local optimization that may prove to be very inefficient in the long run. Worse yet, too many supporting organizations may impede the coordination that is needed if we are to realize a whole that is substantially greater than the sum of its parts.

If a single entity were already aiding CSS projects with coordination, these challenges would all be more manageable. If that same entity were also helping to educate the higher education community about SOA, then it would be ideally positioned to make intelligent decisions about how best to advance projects like MESA. The Sakai Foundation and the

Kuali Foundation are both doing an excellent job of orchestrating their respective projects (indeed, Kuali is already the home for several related efforts); however, both are also constructed by charter with missions that are domain specific rather than focused on the needs of higher education broadly. It may be time to step back and look at the bigger picture, to see if higher education can simultaneously improve efficiency while creating a new technology infrastructure that will serve our needs now and well into the future.

Endnote

1. See, for example, Bradley C. Wheeler, “Open Source 2010: Reflections on 2007,” *EDUCAUSE Review* (January/February 2007): 48–67, or Christopher J. Mackie, “Open Source in Open Education: Promises and Challenges,” in *Opening Up Education: The Collective Advancement of Education Through Open Technology, Open Content, and Open Knowledge*, ed. Toru Iiyoshi and Vijay Singh (Cambridge, MA: MIT Press, forthcoming).