Open for Social: How Open Source Software for E-Learning can take a turn to the Social.

James Laffey  
Matthew Schmidt  
*University of Missouri, USA*

Christopher Amelung  
*Yale University, USA*

**ABSTRACT**

Online learning in K-12 and higher education has been growing rapidly, and open source software has the potential to improve the quality of e-learning. This chapter describes how FOSS enables turning e-learning from a potentially restrictive and narrow framing of the education experience to an emergent and social experience. The chapter will identify several key elements of the FOSS model that position open source initiatives to contribute to the emergent and social nature of experience in e-learning. The chapter also describes several challenges to developing FOSS in a community of educators for e-learning. These elements and challenges are illustrated in a brief case report about the development of an open source software system called Context-aware Activity Notification System (CANS). CANS (http://cansaware.com) is a notification system that integrates with collaborative work and learning systems and is designed around the importance of awareness of user activity, a user’s social context and personal notification preferences.

**KEY TERMS**

e-learning, FOSS, notification system, course management system, activity awareness, social ability

**INTRODUCTION**

Nearly 4 million students were enrolled in at least one online higher education course in 2007 (Allen & Seaman, 2008). This number represents a 12% increase over the previous year. In addition to online courses, web-facilitated courses and blended courses (courses that meet face-to-face but have portions of content delivered online) are spreading across campuses (van Rooij, 2009). As an example, at the University of Missouri-Columbia (MU), thousands of courses—representing tens of thousands of student enrollments—use online course management tools each academic year. In the Fall 2005 semester, 83% of all MU students enrolled in at least one course that blended face-to-face instruction with online learning through the use of a course management system (CMS).

While lagging somewhat behind the growth in higher education, online education in K-12 is also growing rapidly. Based on a national survey of school district administrators, the Sloan Consortium estimates that over one million K-12 students were engaged in online learning in
the 2007-8 school year (Picciano & Seaman, 2008). This number represents a 47% increase from the 2005-6 school year. The report recognizes a special value of online courses for small rural districts in that online courses can provide access to courses that cannot be made available because of teacher shortages or other limiting factors. However, the range of students opting for online learning spans those seeking Advanced Placement courses, those with special needs, those choosing home schooling, those needing credit recovery and those who for some reason do not fit well in traditional school settings. Clayton Christensen, author of “Disrupting Class” (Christensen, Horn & Johnson, 2008), predicts that by 2013 10% of all K-12 school enrollments (with approximately 53 million K-12 students in the US) will be online and that by 2018 the number will be 50% of all enrollments. As another indicator of growth, the fifth annual “Keeping Pace Report” (Watson, Gemin & Ryan, 2008) notes that, as of Fall 2008, 44 states offer significant online learning opportunities for students with 34 states providing state-led programs.

Online courses have great potential to improve access to education. Positive reports of online learning success show its impact and potential, such as relative equivalence in test-result outcomes with face-to-face learning courses (Talent-Runnels, Thomas, Lan, Cooper, Ahem, Shaw & Xiaoming, 2006). However, while online learning is far advanced from traditional correspondence courses, concerns remain about a diminished social experience in online courses that may be detrimental to both the student-to-instructor and student-to-student relationships needed for sufficient engagement and retention of online students. (e.g., Yang & Cornelius, 2005; Berge, 2001; Bower, 2001; Hara & Kling, 2000). Indeed, this lack of social interaction was found to be a factor that depresses student satisfaction in online learning (Arbaugh, 2000). Consequently, dissatisfaction with online learning may be seen in high rates of attrition of online students. Chyung (2001) found that online learners who dropped out perceived that their online learning environment was not engaging, had low levels of confidence while learning at a distance and had low satisfaction levels for the instructional processes used in the online learning environment.

Today’s approach to online learning is implemented through CMSs, of which proprietary systems, such as Blackboard and Desire2Learn, and open source systems, such as Moodle (http://moodle.org/) and Sakai (http://sakaiproject.org/), represent popular applications. These CMSs implement ways for instructors to give and control access to information about a course (syllabus, assignments, grades) and about the subject matter (instructional resources). They also provide some facilities for direct interaction through discussion boards and chat rooms. Such approaches help manage the course and the information of the course but are very limited in how they support the interaction, coordination and cooperation needed to do course activity. Students today use Twitter, YouTube, Facebook and many other social networking applications (Dunlap & Lowenthal, 2009) and are accustomed to their online experience being a social experience (Caruso & Salaway, 2007). Given that a large portion of students use social networking sites daily (Caruso & Salaway, 2007), it is easy to envision many of these students multitasking with Skype and Facebook and other applications while working in their CMS. Indeed, a qualitative study (Goggins, Laffey & Tsai, 2007) of how members of small groups cooperate in online courses found that multitasking while doing online coursework was a key theme. Multitasking occurred in settings where the student was using mobile devices to engage in coursework and when the user was physically present with non-class others while doing the online coursework. We argue that this adaptation by the student leads to unequal resources for students and to unmanaged and unsupported experiences for students. In an era where the need to understand and design for “human computer interaction” is being replaced by the need to understand and design for “human-to-human interaction via the computer” and where e-learning needs to fully mediate teaching and learning, not simply support face-to-face
instruction, current CMSs represent tired and limited ways of working (Vrasidas, 2004; Marra & Jonassen, 2002).

So what does it mean to have CMSs take a turn to the social? One key aspect of being social is to have a sense of community. Sense of community is an attribute of being social that represents a feeling of belonging and having others to ask for support (Blanchard, 2000; Wellman & Gulia, 1999). Having a high sense of community indicates a greater flow of information among members, availability of supports, commitment to group goals and higher collaboration among members (Wellman, 1999; Dede, 1996; Scott, 2004). Sense of community and the ability to interact effectively with others have been identified as two critical factors influencing students’ level of online participation and social interaction (Rovai, 2002; Putnam, 2000; Lin, Lin, Liu, Huang, et al., 2006; Picciano, 2002). Students in their non-academic online experience are finding this sense of community through Facebook, Twitter, blogging and other forms of social computing, but these social networking capabilities are not found in current CMSs.

A second key aspect of being social is activity awareness. By activity awareness we mean being aware of what significant others are doing around you and knowing that these others are also aware of what you are doing. Dourish and Bellotti (1992) defined this form of awareness as "an understanding of the activities of others, which provides a context for your own activity" (p. 107). Carroll and colleagues (2003) identified three different types of awareness information for productive synchronous and asynchronous collaboration: social awareness (who is around?), action awareness (what is happening to objects?) and activity awareness (how are things going on?). They noted that a core problem of the educational site was the complexity involved in maintaining awareness for long-term activities. Traditional classroom-based learning includes natural forms of awareness because members can watch their teacher, fellow students and changes to classroom objects during and across class periods. There is usually little reason for activity awareness outside of the classroom because homework assignments are typically crafted to be individual tasks and instructors do not presume that students work together on their homework. Online learning throws this model on its head as natural forms of activity awareness are no longer available for teaching, and learning activity and non-instructor led collaboration or at least connectivity is usually built into the online learning tools. The opportunity for collaboration and learning with and from others online can be supported by activity awareness, but to make activity awareness effective we need to understand what knowledge about the activity of others is valued, how members relate to each other and use activity information in forming and shaping those relationships and how CMS can represent the needed activity information in ways that resonate with how people work and learn. Carroll and his colleagues (2003) concluded that to coordinate and effectively work in the online cooperative world of educational settings, users (instructors and students) need sets of tools for managing objects in the learning process, knowing when someone does something to an object and keeping track of objects over a span of time and work practice. Their work suggests the need for notification systems that not only convey information, but also put that information into the context of work practices and social roles.

The current CMSs have a large installed base in education, and educational institutions are notoriously slow to adopt new technology. Thus, those working with CMS systems need ways to incorporate new, open and social approaches to supporting teaching and learning. We believe that access to code, an ethos of users working collectively to meet their own needs and community support for innovation make FOSS in education a key approach for helping e-learning take a turn to the social. In the following sections we forward a justification for these claims using a brief case report of our experience participating in FOSS development...
with the CANS project to illustrate our progress and key lessons learned.

**FOSS AND POTENTIAL FOR E-LEARNING**

It is probably a mistake to think of FOSS as a single idea or approach. FOSS is dynamic and evolving in who, what, how and to what impact. FOSS is an approach to software development and distribution that includes source code and forms of licensing which permits ready customization and evolution while preserving the software as a common good. FOSS is also a system of development with demonstrated advantages for reliability, security and adaptability over closed source or proprietary software. FOSS represents an ethos as well as sets of communities with common goals and a commitment to collective invention. The FOSS approach started with early time-sharing systems at MIT and with the development of the ARPANET in the 1960s. In those days FOSS was simply a smart way of working that supported systems improvement, the work of integrating new systems with existing systems and learning how to harness the power of computing systems. In the 1980s, in response to the growing presence and dominance of proprietary software, a "free software" movement began in order to create special licensing approaches to ensure open access to software. For 20 years or so this movement was a vibrant but, for most practical educational purposes, a fringe approach to software development. However, in the last 10 years open source has captured an increasing share of software development because users see it as an attractive alternative approach to the dependence on a single vendor created by proprietary software.

We view the value of open source software for the improvement of CMSs as hearkening back to the roots of the FOSS movement when “freedom” meant flexibility and empowering the union of developer and user, and when “community” did not just mean customer, it meant an environment and network in which your own practices took place. FOSS provides opportunities for stakeholders, from users to developers, to participate in the community development effort which simultaneously contributes to meeting local needs (Lin & Zini, 2008; Carmichael & Honour, 2002). However, it is important to note that participation in and coordination of FOSS community development requires dedicated resources and expertise. While there is little doubt that many universities employ tremendously talented software developers, many of them are students who move on to work in industry and other jobs upon matriculation. At the same time, applying limited development resources to work within the greater FOSS development ecosystem allows universities to leverage the resources of the larger community to their individual needs and the needs of their users (Coppola & Neelley, 2004).

Our vision for how open source software can improve the quality of e-learning is to reassert a more direct relationship between development communities and users so that developers are members of the user community, not simply providers, and where innovation is driven by the integration of the practices of teaching and learning with the flexibility and freedom to develop. To date, this vision has not been systematically approached in the literature. While the impact of FOSS on building pedagogically valid learning environments has yet to be assessed (van Rooij, 2007), educational institutions are looking to FOSS as a way to balance pedagogy and administrative efficiency. A starting point for addressing this gap in the literature is to systematically investigate lessons learned from FOSS e-learning communities like those which have emerged around the open source CMSs Sakai and Moodle (van Rooij, 2009). We view our own story (Laffey & Musser, 1996; Musser, Laffey & Lawrence, 2000; Remidez, Laffey & Musser, 2001; Laffey et al., 2003; Laffey & Musser, 2006) of working in higher education over the last 15 years to build open source software to improve education as a case that illustrates both the potential and the challenges inherent in using FOSS to improve
the quality of e-learning and especially the social nature of the e-learning experience. We use a case report about the development of CANS, an activity notification system for Sakai, to share a part of that story and the perspective we have developed through those 15 years of work.

**CASE REPORT: CANS**

CANS is licensed under the Educational Community License (version 1.0) open-source license, and was designed to provide activity awareness, that is, awareness of what is going on in the CMS, for instructors and students. CANS captures activity information by establishing a vocabulary of tools and action events, maintains a history of activity, makes notifications available based on the context of use and allows users to configure their notification preferences. CANS works by observing activity in Sakai, such as when a member logs in, reads a discussion board item, uploads a document, or enters a chat message. The records of all these observations are stored and matched with profiles for access to awareness information set by the members. Matches lead CANS to send information to members who want the information in a form they have selected. For example, a student in a group may want to know when the instructor has posted an assignment and have that information immediately emailed or delivered via a desktop widget (a small application that can always be visible on one’s desktop). The student may want to see who has posted new messages or read existing messages, but only want that information when they enter the course website. An instructor may want the same information but want it organized in a table to see who has contributed and how much to a discussion. Thus, the awareness information is a resource for instructors and students in knowing and understanding what is happening in the course, for making decisions about when and how to act and also as a tool for an instructor to identify to what extent his or her expectations for student behavior are being met. Given this brief introduction, we direct readers who want more information about CANS to [http://cansaware.com](http://cansaware.com) or to a chapter directly about the system (Laffey & Amelung, in press). The remainder of the case report is provided to illustrate the benefits and challenges of developing a system like CANS with a FOSS approach.

**First Phase:**

In the late 1990s a project team at MU developed an open source licensed intranet system for schools called Shadow netWorkspace™ (SNS), intended to help poor and rural schools use network services such as file sharing, discussion boards and email, and do so within a secure environment. We were inspired by seeing how our own work and teaching was changing with Internet services and how slowly many schools were moving to put these services into place. For example, at that time the service provider to schools in Missouri was allocating 2 email addresses to school districts which then had to share those among administrators, teachers and students. As SNS became operative as an online information system we started to use it in our own teaching and project work. We complained amongst ourselves that it took too long to go into a course and look around to see if students had posted in the discussion board or uploaded documents, etc., so we decided to build a monitor that would show what had been recently done. Fortuitously we created the Activity Monitor (see figure 1) with links to the objects they referenced. In our own behavior and in those we observed in usage studies we saw members using prior students’ activity as a menu for linking to the objects. In short, the monitor transformed the interface from one based on system structures to one based on user activity and changed our perspective from developing an information system to developing a social system.

*Figure 1. Illustration of how the Activity Monitor was integrated into the*
These insights led us to focus on the social nature of the online learning experience and develop a construct and instrumentation for assessing social ability in online learning (Laffey, Lin & Lin, 2006; Yang et al., 2006; Lin & Laffey, 2006). By social ability we mean the person’s capacity to associate with fellows and to use the members, resources and tools of the social context to achieve something of value. However, the implementation of the activity monitor in SNS proved to be problematic. As the semester progressed and there were more and more activities to examine, the activity monitor substantially slowed down SNS’ overall performance. The architecture and mechanisms of the monitor needed to be reconsidered and redeveloped. Chris Amelung, one of the doctoral students on our SNS team, took on the redevelopment for his dissertation work and the result was the first implementation of CANS (Amelung, 2005). In 2005 our academic unit also made the decision to retire SNS as our CMS and implement Sakai. Sakai is a collaboration and learning environment used by around 100 institutes of higher education as a CMS. Sakai is a free and open source product that uses a community source development model. CANS was readily re-implemented in Sakai and was being used at MU as early as 2006.

This preliminary work and early phase of the CANS project illustrates several advantages of FOSS in e-learning. First we identified several key insights about information systems through a “continuous improvement” process that put our own ideas into action. Educational researchers also use the term “design research” as a methodology for continuous and simultaneous efforts to improve both systems and theory (Laffey, Amelung & Goggins, 2009; DBRC, 2002). Secondly, access to code and the ability to create a local modification of a system allowed Amelung to first develop CANS for SNS and then port it to Sakai. And, thirdly, we were quickly able to continue our implementation and research into the social nature of learning because members of the Sakai community provided expertise and consultation for making the move to Sakai. However, FOSS also has some downsides and one illustrated in our report is the shuttering of the SNS project. Because the K-12 schools for whom it was intended, primarily rural and poor, did not have many resources to put to its implementation and not any to put to its continuous improvement, the weight of advancing
SNS, even to upgrade for new versions of browsers, fell to a small team of developers and eventually the system could not be sustained. FOSS projects need to develop communities of supporters and contributors in order to thrive and survive.

**Second Phase**

Shortly after the move from SNS to Sakai we were granted an award from the Fund for the Improvement of Postsecondary Education, U.S. Department of Education (Project # P116B060045). The grant allowed us to continue the development of CANS to include a listener for Sakai and new forms of notification. The listener works within Sakai to report events (discussion posts, downloading a file, opening an assignment document, etc.) to the CANS server. One challenge we encountered was the diversity of ways that events were represented across various functions of Sakai. Because Sakai is a collection of programs from various universities integrated by some core services, it offers a great breadth of e-learning functionality but also much diversity in how each application is programmed. We also developed three ways of reporting events back to the members: an email digest, an interactive webpage and widgets. The digest provides a reporting mechanism that is outside of Sakai so we do not have to depend on students logging in to Sakai to be informed. The interactive webpage allows the user in Sakai to directly link to a web page with capabilities for dynamically (using FLEX technology) representing social information allowing for when members are engaged in the learning activities and have a question about the levels and types of activity underway in the class. The widgets provide flexibility to deliver information directly to a member’s desktop or can be embedded in the homepage of the course. Figure 2 shows an example of a widget providing social comparison information so a student knows how their activity in certain aspects of the course compares with that of the most active student and the average for all students. The social comparison widget may have less detail about what has been done than a digest, but in this example the widget frames the information so as to make it personally relevant to the question of, “How does my level of activity relate to that of others in the class?”

*Figure 2. A sample social comparison widget*
As part of our work in developing the new notification features we undertook a case study of a group working on a collaborative project within a class. Findings from this study showed some of the challenges of being an active member of a social learning unit when computers are used to fully mediate learning. Two core themes came from the analysis of how the group worked. One theme was an emphasis on managing social identity so as to maintain a cordial environment and not risk social capital. Members stated that they did not disagree with people online because it is too easy to have things misunderstood or result in bad feelings and, after all, they did have to work with these people over an extended period of time and a large number of activities. Such findings provide evidence that the available tools of current CMSs do not provide appropriate mechanisms and structure for argumentation and fail to provide the contextual cues for managing those types of exchanges. In fact “thin communication” such as that which we try to achieve with activity awareness may be viewed suspiciously by participants. By thin communication we mean the abstracting of certain attributes of an activity (such as it happened) and providing that in a list or other de-contextualized form. What makes this type of communication somewhat problematic is that instead of a person making the observation that someone just looked at their report, they may wonder “why” someone looked at their report. In our theoretical world of social navigation we might predict that seeing someone look at a report might cause others to look at a report. However, in the real world it might create suspicions of copying ideas, or feelings that others were checking up on you, etc. The other theme is one of multitasking. Students are doing their coursework as a part of their other activities. For example respondents told us, “I’ve actually been driving down (the highway), talking to my husband and posting to my team,” and “Sometimes I get distracted when my boyfriend blows something up in Gears of War while I’m chatting with my team online.”

Awareness as a support for participants in an environment where they are likely to be
multitasking and with many possible distractions from the online learning tasks became an important attribute of the users’ design personas. As we considered our design in the context of multi-tasking learners, one approach was to consider how to focus attention on the learning tasks and mitigate distractions. Further analysis drew us to the belief that we needed to embrace the idea that students were multitasking and support how they managed the learning tasks within their multitasking context. They were multitasking across course tasks, across courses and outside course tasks. For some students it appears that multitasking is a natural way to work and learn, for some it is necessary given timelines and responsibilities and for others the stimulation of multiple activities or distractions seems to be an alternative to falling asleep at the keyboard.

To address the issue of multitasking we conceptualized two dimensions that seem to be in play for awareness information in multitasking situations. One dimension is salience. Awareness reports may be too meek to draw attention given other factors being considered, such as in the examples mentioned above of driving and observing a game being played. The opposite end of the salience spectrum may be overbearing intrusions that, especially in a multitasking situation, could lead to awareness overload. A second dimension is meaningfulness. How will users make sense of the new information? If the information is too disconnected from the users’ mental models it will be interpreted as noise and disregarded. If the information is tightly coupled to a specific event or process, it may be represented in a way that is inefficient. For example, if a user is developing a response to a discussion board message, it may be helpful to know that the author is currently online and working in the discussion board tool, but overwhelming to also know details about the new thread the author is in and activity surrounding that thread. This nuance seems similar to the construct of social translucence (Erickson & Kellogg, 2003). Social translucence refers to the goal of providing appropriate information and representations so that human-computer interaction (HCI) can be gracefully undertaken. A physical example is when two people approach a door and one may open the door into the face of the other. Having a glass door allows for seeing the other, but the needed action does not actually require knowing who the other person is or distinct attributes about them. Thus the door can have a filmed glass so that all that is known is that another person is approaching. Another aspect of social translucence is accountability. It may be excusable to open a door into someone’s face if a wooden door made it impossible to know the other person was on the other side, but once cues are available for “knowing,” then you are accountable for your actions. This objective of providing activity awareness information as needed and in appropriate representations seems a good fit for helping members derive meaning in a multitasking situation and creating online learning environments where users are more accountable for their actions.

This recognition of the importance of social identity and multitasking and the complexity of providing salience and meaningfulness led us to design a notification manager application in Sakai that would allow instructors or students to set up what, when and how they wanted notifications. The manager application allowed us to not only provide activity and social information in multiple forms, but also to allow those forms to be customized by local knowledge of the learning tasks and how members related to each other. For example, a course that used team project work could set up digests or widgets for the team members and exclude other members of the course from being able to access this information. This might allow members to be more comfortable with what type of information was shared and to have it be more meaningful to the decisions they were making about how to work. However, the step of building a notification manager that could be accessed as a Sakai tool required a programmer with experience developing for Sakai to ensure that the interface and the way data were handled conformed to best practices in Sakai programming.
The work we undertook in phase 2 of developing functionality for CANS to work within Sakai showed the power of integrating our work with a larger community, in that Sakai had many more functions for e-learning than did SNS. Clearly, our work also benefited by our ability to add functionality to CANS, such as the notification manager, as we developed new insights about how activity and social information is used in e-learning. This ability for local adaptation as a means to test approaches to supporting the social nature of online learning and to investigate the impact of the approaches, and then to repeat the process based on newly informed ideas, is needed for continuous improvement and knowledge building. This ability for design research and continuous improvement of software and testing in large systems deployed at educational institutions is uniquely supported by the FOSS model. Just imagine what it would be like to try and talk your campus administrators and the executives of a proprietary system to do a trial of your new idea! However, this ability to try innovative approaches may not be easy to execute. We found substantial complexity and overhead in trying to integrate CANS within Sakai. The need for a programmer with capabilities specific to implementation in Sakai could not be readily filled by our team, but fortunately was available within the larger community.

**Third Phase**

In the third phase of the CANS project we are attempting to support other universities in the use of CANS. At present we are focusing on efforts to test the viability and usefulness of CANS at the University of Michigan, Rice University and East China Normal University (ECNU) in Shanghai, China. All thee of the universities are performing some form of testing with a plan for a pilot implementation. This step beyond MU shows some key benefits of working with FOSS and some key challenges. On the benefit side, it is an exciting part of the design and development process to find collaborators who share your vision and are willing to invest time and staff to try the software as well as make contributions that extend your own work. These other universities bring experience and expertise that go beyond what our team at MU can provide. For example, the University of Michigan operates a large network to meet the needs of its very large Sakai user base. Heretofore in our development we had not addressed the need for load balancing and other concerns to support system performance under high load conditions. Their leaders and staff address these concerns as a common part of how they work and are able to both identify issues and help us address them. The leaders at Rice University have particular interests in using data for institutional research as part of their academic computing decision-making. Their concerns and insights are both pushing us and helping us see new approaches to extend the value of CANS beyond notification in the learning context. Similarly, at ECNU they need a version of CANS that can report in Chinese and are developing solutions to make language a variable within the system. All of this collaboration extends our small team and pushes and pulls CANS to be a better system.

A motto of the Sakai community is “Designed by Educators, for Educators.” This motto implies that there may be special insights that those who practice education can understand and bring to the design work that may be hard to identify and articulate in a product development cycle that includes educators as customers and not as a community of practitioners. This is not to say that FOSS is the only or even best way to develop software for education, but just that it is a different way than developers of proprietary systems apply and it has some unique benefits. We characterize a key difference in development approaches as proprietary companies have a great focus on locking in the value of their accomplishments, by adding to their customer base and by seeking patents on their inventions. In contrast we see the focus in the small CANS and larger Sakai community as unlocking the potential of the technology to meet felt needs. In part, this unlocking comes from the daily experience of applying the technology, confronting its limitations and seeing its possibilities. Of course, for
long-term success one must be able to both lock in value and unlock potential. Our view, that was expressed in the introduction about the limitations of CMSs, is that at this point in time the need to unlock potential is great. Working with partners and being a part of a community is both inspiring and productive for extending CANS, undertaking design research and seeing potential with new eyes.

While earlier we pointed out that diversity of programming creates a challenge for integrating CANS into Sakai, diversity of thought is a key asset of working in a community. In the case of Sakai and the goal of making it a more social place for learning, we are developing CANS to impact access to and the use of social information. Other researchers, also having the goal of making Sakai a more social place for teaching and learning, are developing other approaches. A key initiative in the development of the next generation of Sakai is to include social networking functionality. The addition of social networking in Sakai will use Apache Shindig (an application intended to support social networking developed by the Apache open source project). The addition of social networking to Sakai is intended to support collaboration within sites and possibly across sites. Additionally it is hoped that this work will help define the purpose and role for social networking in an academic context.

Phase 3 has also shown us additional challenges for FOSS in e-learning. Working with other universities shows not only the breadth of their needs and interests, but also the diversity of their technology infrastructures. A key attraction to the universities that have adopted Sakai is the ability to customize it to their own local requirements. Perhaps this leads to more diffusion in types of infrastructures found at Sakai campuses than would be found at campuses adopting a proprietary system. We speculate that the decision to purchase a proprietary system starts a process of coordinating all systems to work with the chosen system, whereas a decision to adopt a FOSS starts a process of customizing the FOSS to work with other existing systems. Of course it is also likely to be true that the company selling a proprietary system invests heavily in making sure the system fits with the most popular configurations found on campuses.

The lessons learned through the three phases of CANS development illustrate both the benefits and challenges of the FOSS approach for making online learning more social. Access to source code and support from a community of peers, the fundamental aspects of FOSS, enable local perspectives and expertise to shape new software. In the field of e-learning much of this local experience pushes the software to address the social and emergent nature of learning which is not well supported by current CMS. Progress in this area is further supported by the ability to learn from field tests of new systems capabilities, such as CANS, and improve both the conceptual framework for what you are trying to accomplish and the software systems in use. The challenges of developing software are substantial and while Universities have ready pools of student programmers and resident faculty expertise, this combination is not well suited to the sustained and systematic effort needed to develop software. This challenge is further complicated by the work efforts needed to move from software that works as a prototype to a product that can be distributed to a broader community. Aligned with the challenge of moving from prototype to product is the need to integrate the software with that of the broader community, in our case integrating CANS into Sakai. While our local team had the capabilities to develop CANS we struggled to understand the complexities of integration with Sakai which indeed was a moving target as Sakai is under continuous development to meet other pedagogical needs and administrative needs of the broader community. In many ways it is best for innovation to exist at the margins of a community where new ideas can be tried in a rapid fashion and where failure is appreciated as a learning experience. At some point or points those innovations must be harvested by the broader community to spur
advances and meet needs. This transition has many challenges as the innovation must be ready for product status and the community must embrace change and some level of risk. As van Rooij (2007 & 2009) illustrates in her review of open source implementation case reports, efforts such as ours offer potential for building new knowledge about how open source software can enable improvements to pedagogy and find a balance of sound pedagogy and technical efficiency.

DISCUSSION

Starting from a perspective that online learning, a key form of e-learning in higher education and K-12 education, is rapidly growing and increasingly important in education, this chapter has described a need for improved CMSs to support e-learning with a particular focus on enhancing the social nature of the online experience. We have also characterized the processes needed for supporting the emerging needs of viewing CMS as a social place for learning as requiring innovation, building systems from insights gained from practice and continuous improvement. The chapter has also used our experience of developing CANS to illustrate some of the key benefits and challenges of the FOSS approach and how they fit the new requirements of e-learning. We have identified and illustrated two key themes of how FOSS supports the types of innovation and improvement needed for having CMS take a turn to the social. The first theme is the freedom of development made possible by access to code and licensing forms that allow local modifications. This freedom of development couples with a commingling of developers and practitioners to move from insights for improvement to trials of new code and functions. The second theme is that of having community support for innovation and collective invention. There is a form of natural selection that forms a community around the use and extension of an open source system that encourages and invites innovation and improvement. Having a community means that there are physical, social and intellectual resources that can be brought to bear on challenges.

The theme of freedom to develop was shown in the ability of the CANS developer to have direct access to the code and system representations that stand for activity in a system, such as event messages. This access is enhanced by collegiality among developers in a community and the valuing of innovation. Discussions with other developers led to improved models and approaches to how to utilize events messages. FOSS provides an unprecedented level of software freedom in comparison with proprietary counterparts, which can be both advantageous and challenging. Such freedom allows for profound flexibility in software design and implementation, but it is this flexibility that can result in difficulties like non-standardized event messages. Issues like this are not necessarily problems, but rather the logical outcomes of software freedom. Just as freedom of speech results in wildly diverse opinions, software freedom results in diverse designs and implementations. We argue that diversity is a necessity for influencing and realizing innovation and is core to the nature and success of the FOSS model. However, the freedom that allows for such diversity is not without its own unique set of challenges.

The theme of having community support for innovation is also critical to the work of innovation and improvement in e-learning. For example, the Sakai community has an annual worldwide meeting and various local meetings as well as fostering lively discussions about systems implementations and improvements via listservs. Of course, proprietary systems develop communities of users who meet to discuss practices and lobby for improvements, but lobbying for improvements as a request to a corporate entity is different than being engaged in the discussion about how the improvement will work with the members working to make it happen, or being one of the members making it happen. As mentioned earlier, one can argue
the efficacy of a proprietary or FOSS approach to development for large complex systems. However, we argue that the FOSS approach is uniquely well suited to drive innovation and lead to more social places for learning.

While software freedom and working in a community that support innovation grants flexibility for innovation, this flexibility comes at a price. In the case of open source CMSs, universities and other educational institutions that wish to modify or change the software require resources to do so. Yet, these institutions are often under-resourced. Finding expert developers and programmers to develop open source CMSs is difficult, as these experts are in high-demand in the job market and receive high salaries. And while many brilliant programmers do work with universities, most are students who move to higher-paying jobs after graduation. Hence, significant time and resources are spent continuously training new developers; yet the return on the investment for time and resources spent training these developers is short and unsustainable. This continuing turnover makes it difficult to build strong and efficient development teams. In addition, because many institutions are under-resourced, they must focus on their own specific needs. Consequently, the development done at these institutions may be so specialized that there is limited generalization to the broader needs of the community. And while it is understandable that institutions want to invest their resources to meet their own specific needs, attention to matching these specific needs to the broader needs of the greater community is a challenge for the leadership in any FOSS project. However, it has been our observation that even when the specific code developed at one institution may be limited in its application across the community, the community advances by seeing the efforts and understanding the context of the new development. In this way the community develops both social and intellectual capital from innovative practices and the sharing of ideas and results. A key illustration of this in the Sakai community is the sharing of best pedagogical practices using Sakai across institutions. An awards program highlights faculty members who are recognized for innovative and effective use of Sakai in their teaching. No one expects a faculty member from another institution to simply implement the same course on their campus, but the sharing of information within a community that values innovation is a great asset. This last point is also supported related to software development by our own CANS Phase 3 development story and underscores the importance of community and working with others in FOSS to account for diversity of needs and interests.

Our story of CANS development also illustrates the need for a community of contributors, not just users, in order to sustain a project and the challenge of integrating contributions in complex development environments such as that encountered in Sakai. For example, in Sakai there are many developments and developers. One result of this is that event messages from one application in the system (e.g., a forum tool) may have different expressions than event messages from another tool (e.g., a wiki tool).

To be sure, working within a community and expanding the scope of CANS to encompass the needs of other institutions has helped us to make advances and improvements in the project. From a development perspective, the project has attracted more development input from other institutions, which is helping to grow CANS in scope and impact. Considerations of incorporating CANS in heavy-load environments, using CANS for decision-making and even incorporating multi-lingual support are improvements that result in a more mature product which is better suited to meet the needs of the greater community. In addition, by expanding the scope of CANS we are effectively introducing more use cases to be considered for further design and development work. By incorporating into our design processes the needs of a diversified user base that goes beyond the original constituency of the CANS team, we are able to innovate and push our ideas and theories further. And, by including the user base in the
design and development process, we create a sustained feedback channel, allowing for continuing improvements to design and development work. These factors compounded make it possible for us and others to create software that better aligns with the social needs of online learners and better supports the social nature of online learning.

The work reported in this chapter was partially developed under a grant from the U.S. Department of Education. However, those contents do not necessarily represent the policy of the U.S. Department of Education, and you should not assume endorsement by the Federal Government.

REFERENCES


