AC 2008-1750: PRELIMINARY EXPERIENCE OF USING A LEARNING AND KNOWLEDGE MANAGEMENT SYSTEM FOR AN SE-1 COURSE

J. Scott Hawker, Rochester Institute of Technology

Dr. Hawker is an Assistant Professor of Software Engineering at the Rochester Institute of Technology (RIT). He graduated with a B.S. and M.S. in Electrical Engineering from Texas Tech University. He graduated with a Ph.D. in Electrical Engineering from Lehigh University. He has over 15 years of industry experience developing large-scale, multi-agent information and control systems for diverse applications including manufacturing, combat pilot mission decision support, robotics, and surveillance. In these areas, he developed and applied technologies including distributed, component-based software architectures, software and systems engineering process models, intelligent control, the semantic web, and real-time artificial intelligence. Dr. Hawker has been a faculty member teaching software engineering and computing topics at the University of Alabama and RIT. His work with the NASA Technical Standards Program applying semantic web, formal modeling, information retrieval, and other advanced information technologies inspired his work to better create, manage, find, deliver, and use learning content in software engineering courses and projects. Dr. Hawker is also co-director of the Laboratory for Environmental Computing and Decision Making, which focuses on modeling and understanding the environmental impact of freight transportation and automotive industry activities and public policies. Dr. Hawker is a member of the American Society for Engineering Education, IEEE, IEEE Computer Society, and the ACM.

Ian Webber, Rochester Institute of Technology

Michael Starenko, Rochester Institute of Technology

Jeremiah Parry-Hill, Rochester Institute of Technology

Preliminary Experience of Using a Learning and Knowledge Management System for an SE-1 Course

Abstract

Students and instructors struggle to provide an integrated view of learning content that is spread over personal and shared file systems, course management systems, team project repositories, wikis, blogs, and other content storage and retrieval systems. Further, they struggle to organize this content when each system provides a different organizational mechanism and strategy. Course management systems fragment content by semester, course, and class session. File systems fragment content into hierarchically organized folders and files with only folder and file names to describe the content. Project repositories fragment information into versions, artifact types, and subsystems organized as folders. It is especially a challenge for a student who does not yet have the knowledge for how to organize and describe their learning content.

We have developed an initial version of a learning and knowledge management system and piloted it for an Introduction to Software Engineering course. To hold the common repository of learning content, we used digital library technology. To organize content, we tagged content using subject headings based on the Software Engineering Body of Knowledge (SWEBOK) subject taxonomy. Instructors and students can contribute content, tag it using SWEBOK and other terms, and search through it using SWEBOK and other terms.

The results of our pilot were mixed. A digital library-based repository is insufficient for dynamically changing and evolving content. A significant amount of learning content needs to be provided (more than for just one course), and there needs to be one location of record for accessing content, rather than multiple locations. A positive result was some validation that using SWEBOK to organize, tag, and search for content is helpful in readily accessing information and helping provide students with an understanding of the organization of knowledge of their discipline. Since the main intent of this research project was to gain some operational experience and initial validation of using a domain-specific taxonomy to organize learning content, we consider the project to be a success.

Introduction

In prior work[1] we described the need for and design of a knowledge management system that holds learning content for Software Engineering courses, and provides students, project teams, and instructors with advanced tools to create, share, and annotate both the learning content and an organizational structure for that content. Traditional course management systems, team project repositories, wikis, etc., usually fragment information into silos (that is, into distinct information storage locations which are not integrated). We seek a system to help integrate the fragmented information into a whole across the curriculum and the student's academic career and to improve student interaction with learning content and with each other in project teams. Further, by organizing the content and presenting it in a way that reflects the structure of knowledge in the software engineering discipline, we expect that the students will more readily grasp discipline-specific concepts and concept relationships, thereby better organizing and integrating their learning from course to course over their academic and professional career. The intended result is a repository of readily accessible and understandable learning content and a location (a “studio”) for team collaboration.

We have implemented a testbed, called Knowdio (a contraction of Knowledge Studio), that implements a portion of our system vision. The testbed allows experimental assessment of the effectiveness of this learning and knowledge management system. This paper provides a background of our vision and then presents our current system implementation, our initial experiments and results, and planned next steps.

Background - Problem

Students and instructors are challenged to manage course content and integrate it across the curriculum. For example, a student might take five courses a semester over eight semesters – 40 courses. Especially for the courses in the student’s major, the content of these courses are related to content of previous courses, building on and integrating prior learning. Further, within a given course, there are numerous lectures, exercises, exams, and projects that interrelate. When a student gets to their senior capstone design experience, they need to draw upon all that they have learned, combined with all that their teammates have learned, to quickly and effectively develop a quality solution to a real-world problem.

Similarly, an instructor may teach five courses a year over many years, each with dozens of students. Course content changes from year to year, and a given course’s content must be continually aligned with the content of other courses and content from other instructors. As the curriculum evolves and teaching improves, faculty members need an integrated view of learning content and student performance assessment over the entire curriculum and numerous years.

Current approaches to managing course information do not provide the integrated, organized, evolving view of knowledge that students and instructors need. Instead, current learning and knowledge management solutions tend to fragment information into information silos (distinct information storage locations which are not integrated) for each course, topic, semester, project, and team. The silos use different technologies, such as course management systems, version control repositories, personal and shared files systems, ePortfolios, wikis, blogs, and web sites, which cannot easily be integrated for cross-reference and uniform organization and search. Each course, instructor, student, and project team has their own systems and silos. External sources, such as publications, standards, electronic books, etc., can only be integrated via web links and search engines. This fragmentation of learning content is bad enough for a given individual. It is compounded when the individuals need to collaborate in teams and committees!

Each information management technology also tends to limit the ways students and instructors can organize their information. Course management systems organize by semester, course, and class session. Project repositories organize by files and folders named to reflect artifact type (requirements, design, implementation, test, plan, metrics, etc.) and subsystem. File systems organize by hierarchical folder structures where folder and file names are often poor descriptions of the type of content. Blogs organize by time and, perhaps, topic. Discussion boards organize by topic thread. This diversity of organization styles may be manageable by knowledgeable individuals, but students new to their discipline of study do not yet have the terms and concepts to organize their work. For example, a new software engineering student may not know how requirements relate to acceptance tests or how designs relate to unit and integration tests, let alone how to use the given content management technology's organizing metaphor (named hierarchies, web links, etc.) to relate the parts of learning content and project artifacts that deal with those requirements, designs, and tests. Students do not yet have the knowledge of the discipline to provide a good organizational structure of their knowledge, and their information silos do not provide the tools to capture and integrate the multi-dimensional structure of their learning.

Possible Solution – The Knowdio Vision

We envision Knowdio as a learning and knowledge management system that stores software engineering learning content and allows students and instructors to collaborate to create, discuss, and contribute new content. In addition, we envision Knowdio as having powerful tools that instructors, students, and teams use to organize, manage, and interact with content. Through the process of creating, discussing, annotating, organizing, and integrating content, the individuals will interact and collaborate with each other. Further, individuals and teams can interact around the organizational structures to create and share new ways to organize and manage content.

Figure 1 illustrates the scope of our vision. On the left side of that figure is a repository of personal and shared learning content. On the right side of the figure are metadata that describes the content (title, authors, subject headings, description, provenance, access and use rights, relationships among content, etc.) and a taxonomy capturing the organization of the subject headings and relationships among the concepts represented by the subject headings. The content and organization are built on an infrastructure of distributed storage and access control. We suggest that by providing high-quality learning content that is accessible, organized, and searchable using the concepts of the discipline, we will improve student learning and instructor effectiveness. Even though there is some evidence that this approach is useful,[2,3,4,5,6] this is a hypothesis that must be tested in classroom use. In particular, we must assess whether the additional effort to organize, manage, and make the content sharable pays off in improved information access and understanding, and that it helps rather than negatively impacting student learning and instructor effectiveness.

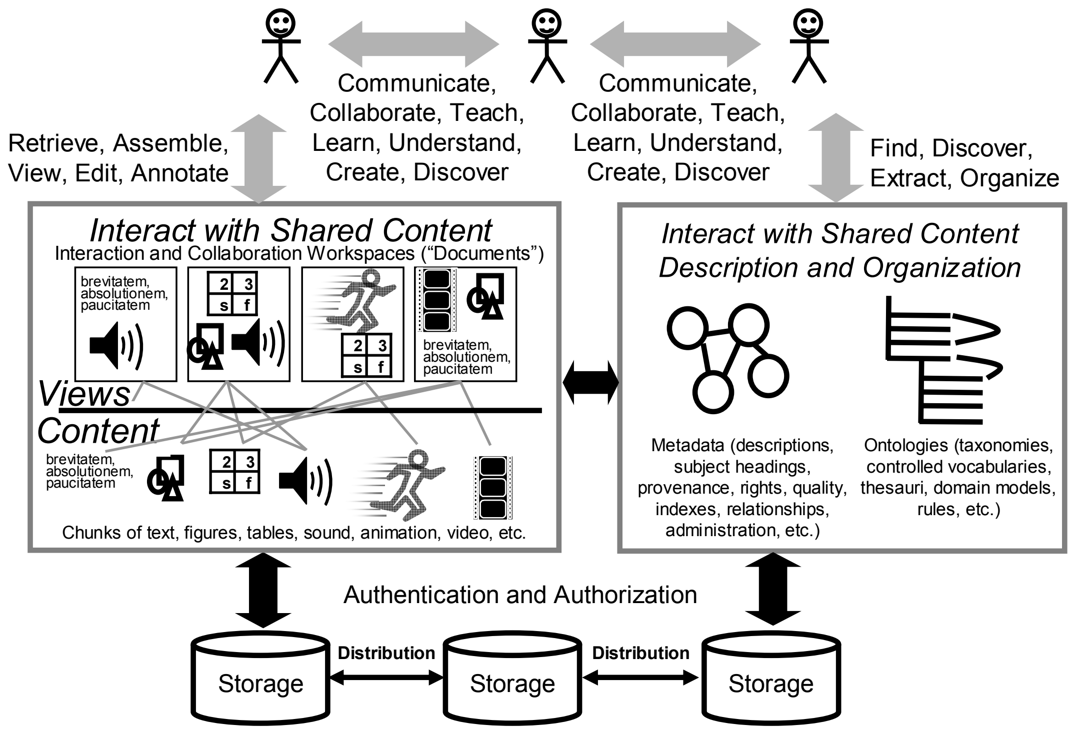

Figure 1. The Knowdio vision[1]

Knowdio Version 1

Figure 2 summarizes the functional scope of a complete Knowdio system.[1] Although open source and commercial technology elements exist for most of this functionality, the effort to select, adapt, and integrate these elements would be rather large. Before embarking on this development effort, we need to assess the validity of our concepts: that the investment in organizing and sharing information across the curriculum pays off in improved learning and instructor effectiveness.

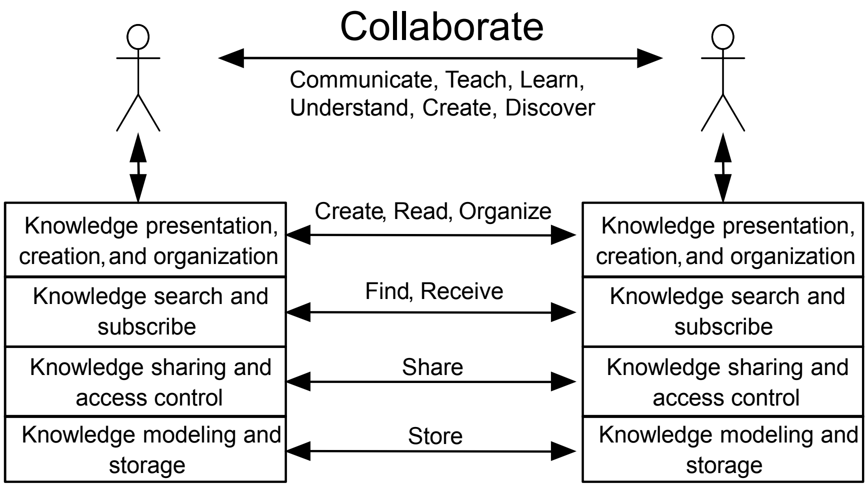

Figure 2. Architecture layers of the Knowdio vision[1]

Figure 3 illustrates the user mental model we are trying to achieve in Knowdio testbed Version 1. All content is organized and accessible through a taxonomy of subject headings that reflect the organization of knowledge in the software engineering discipline. Figure 4 shows sample search results, and Figure 5 shows details of the metadata for a given item returned by the search.

Through a grant and support from our university's Online Learning Department Emerging Technology Partnership Program, we developed and piloted an initial version of the Knowdio testbed. This Knowdio Version 1 is based on the DSpace digital repository product.[7] To organize the repository content, we use content subject labeling, organization, and search terms based on the Software Engineering Body of Knowledge (SWEBOK) subject taxonomy[8] and Dublin Core Metadata[9] (subject, author, date, title, description, etc.). We populated the repository with the learning content of our SE-1 course, including lecture slides, software development process descriptions, artifact templates and examples, and assessment rubrics. Each content item is labeled ("tagged") with SWEBOK terms, thus organizing the content into a SWEBOK-based subject taxonomy.

Students, project teams, and instructors can all contribute content and tag content. All submitted content is automatically indexed for search, and the search user interface offers SWEBOK, Dublin Core, and user-added terms as suggestions for improved search and retrieval. The Knowdio search feature allows users to search for content using these SWEBOK subject terms and any other terms that the user wants, finding content that was either tagged with those terms or has those terms within the artifact (as in a full-text search engine). To enable detailed usage analysis, we instrumented the Knowdio implementation to capture usage: session time, navigation sequence, search terms, search results, etc.

Our initial results using Knowdio Version 1 are mixed: some good experiences and some concerns. The next section describes our Knowdio pilot evaluation, followed by our results.

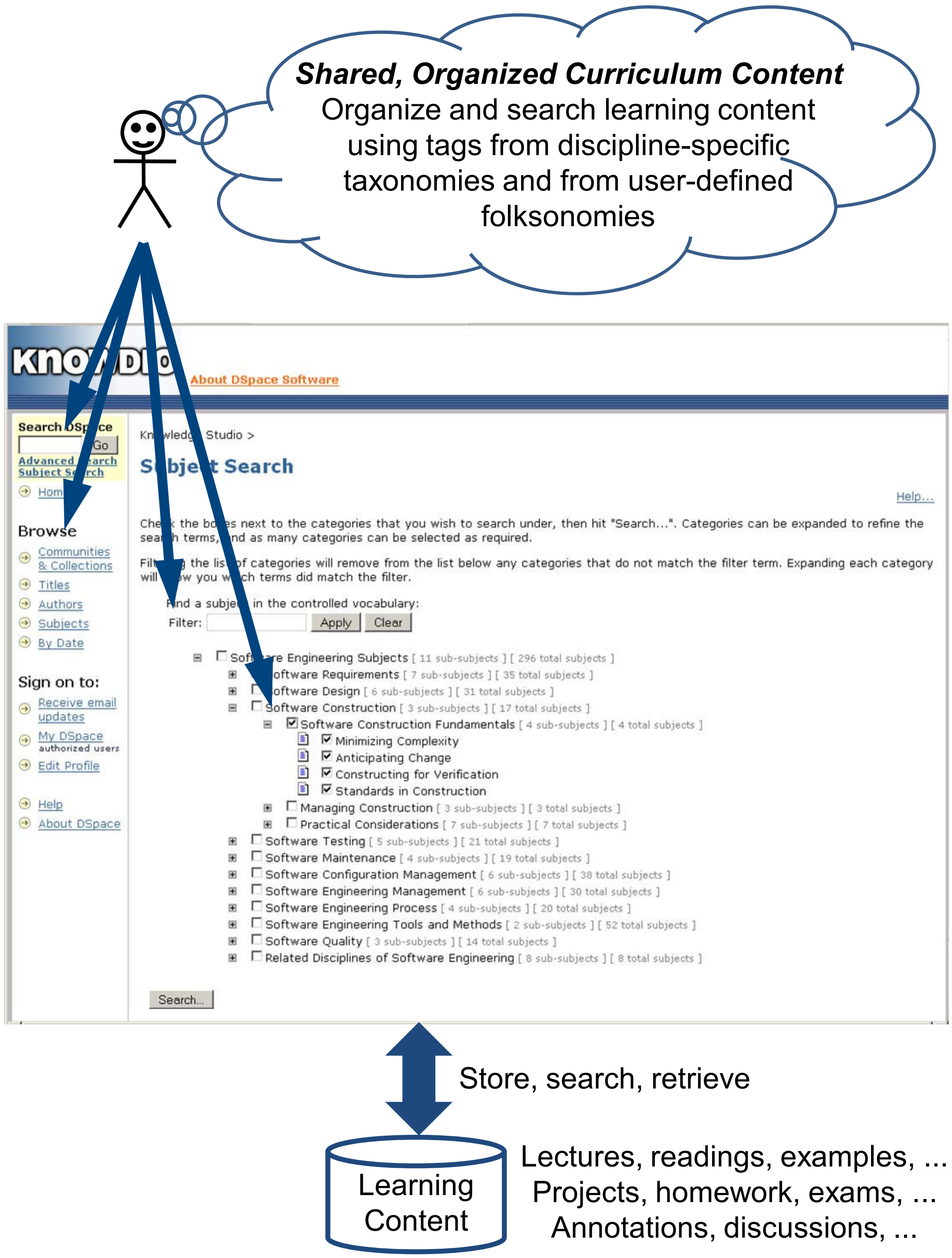

Figure 3. User mental model

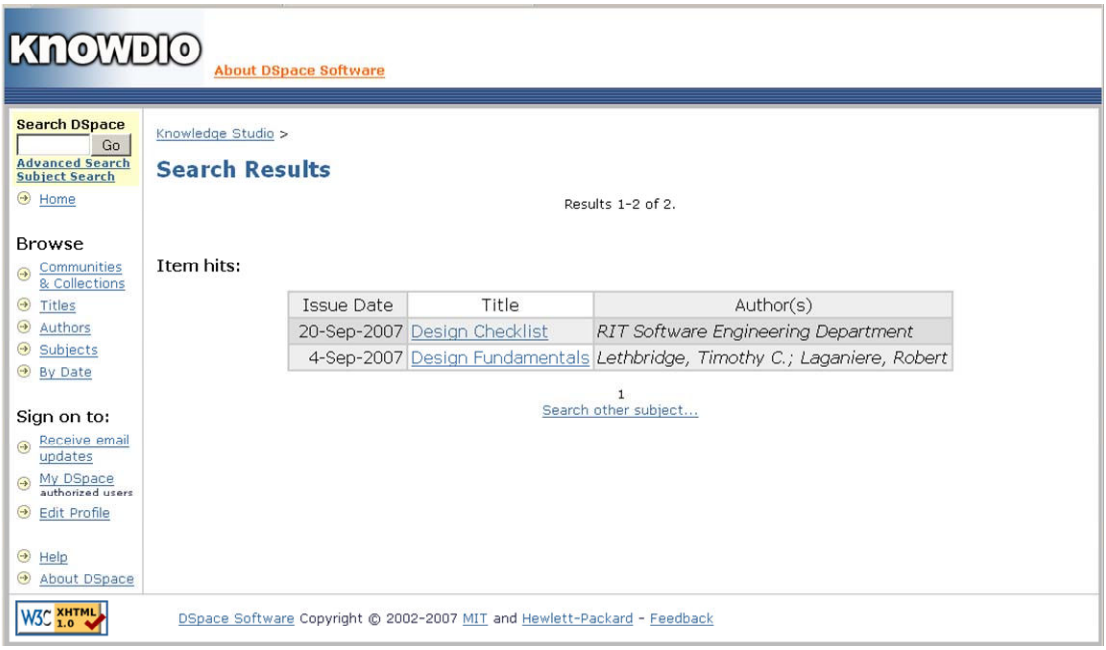

Figure 4. Search results

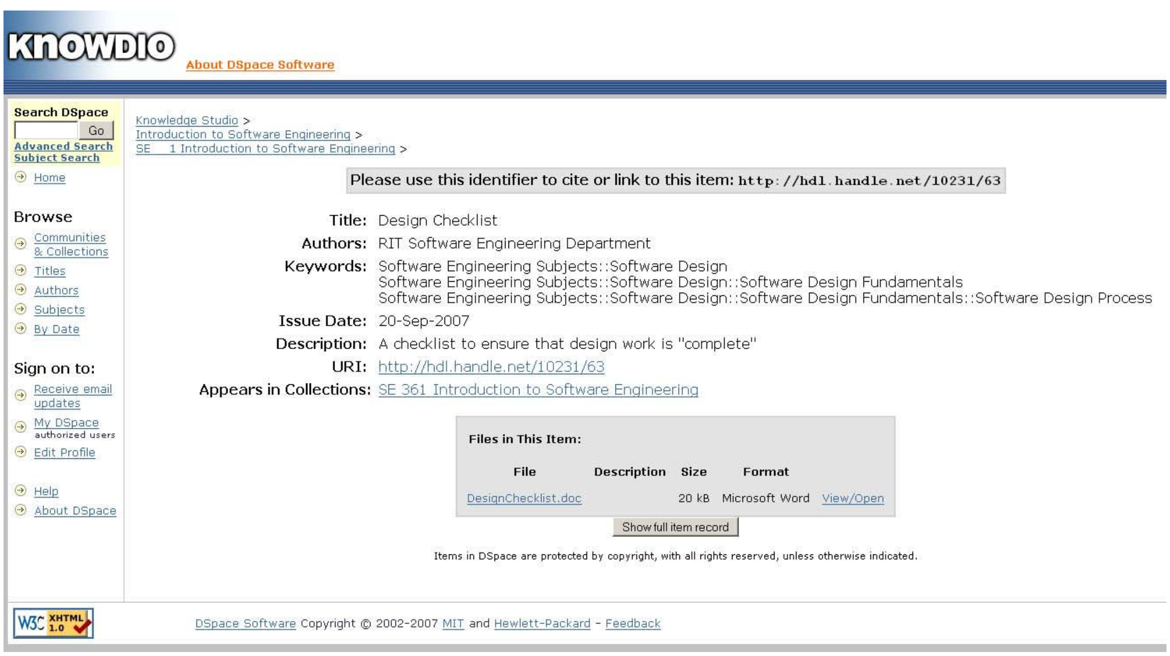

Figure 5. Details on an item in Knowdio

Knowdio Pilot

We piloted Knowdio Version 1 in four SE-1 course sections, taught by different instructors but with common course content and teaching approach. The Knowdio researchers did not teach any of the sections, but the Knowdio principle investigator has taught the course numerous times. Our intent was to have minimal impact on the way the course is normally taught so as not to negatively impact student learning and instructor effectiveness. Knowdio was to provide a supplement to the existing technology and teaching approach. The next subsections describe the current knowledge management technology and content organization for the courses, followed by how we introduced Knowdio and gathered initial Knowdio usage information, and an analysis of results.

Current course knowledge management systems

The course currently uses a course management system based on a commercial product (Desire2Learn[10]) which, as Figure 6 illustrates, organizes content primarily by course session for a given course section and semester. The content in the course management system links to additional content stored and organized in file folders in a web site whose organization is shown in Figure 7. Note that the web site reflects the session-based and year-to-year organization of the course management system, but also has other organizations based on artifact type (articles, activities, forms, rubrics, tutorials, etc.) which is somewhat mapped to the list of items in the view of Figure 6 above the Session Outline content. The student's primary view of instructor-provided course content is through the course management system. They do not see the underlying web site folder structure.

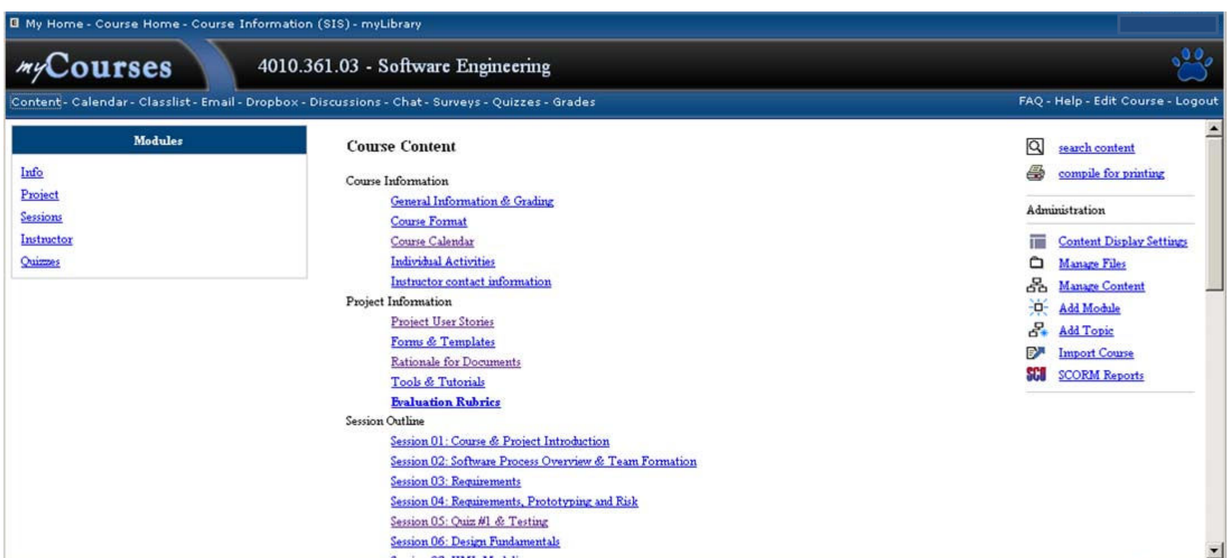

Figure 6. The primary view of the current course management system

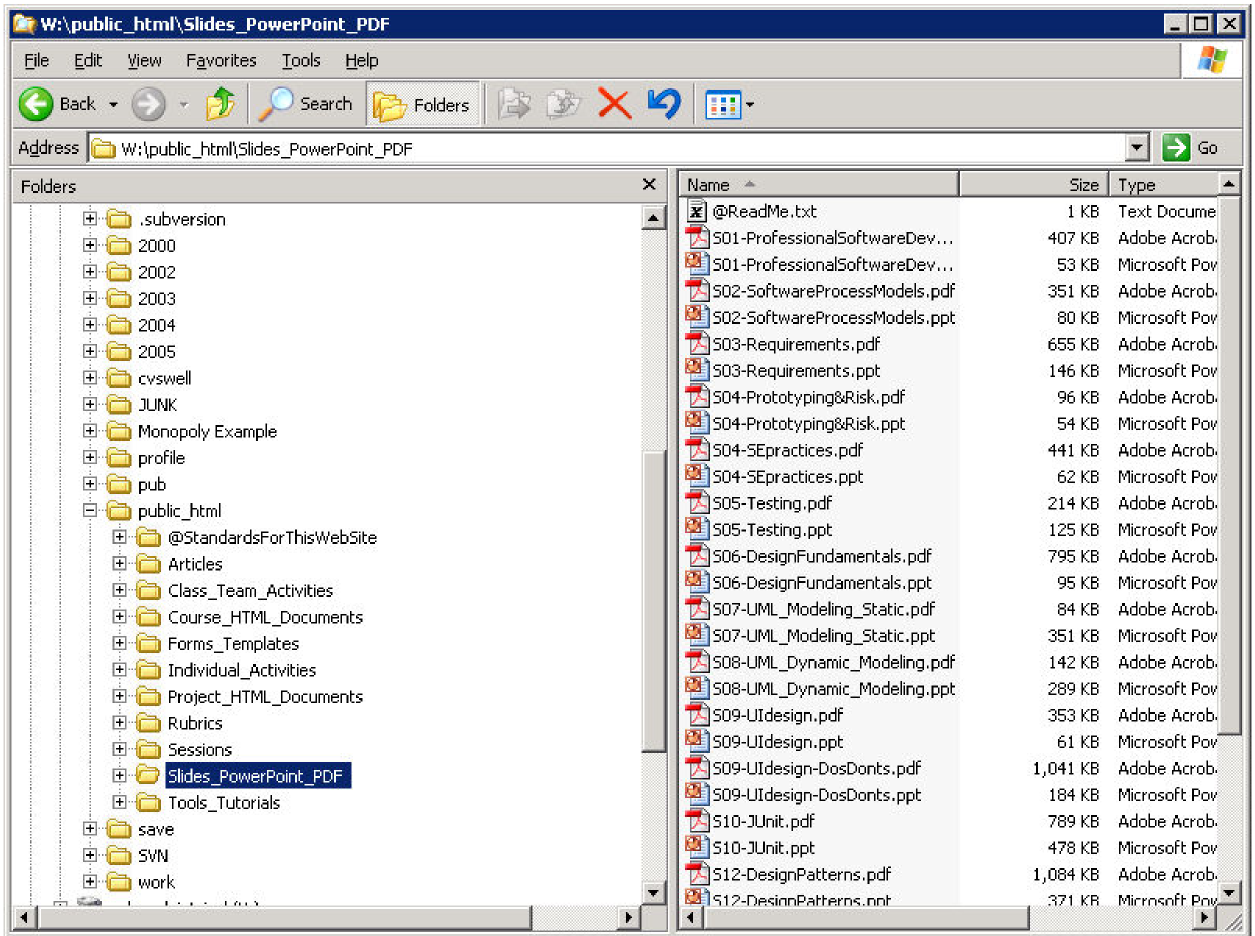

Figure 7. Folder-based organization of course web site

Submission and feedback of a student's individual work is currently captured in yet another structure in the course management system: the "Dropbox" view of Figure 8. Team project work is stored and organized in a version control repository, available through the Eclipse development environment, as Figure 9 shows. Notice that the team project repository provides yet another organization of course content—artifacts, versions, and instructor feedback on project work.

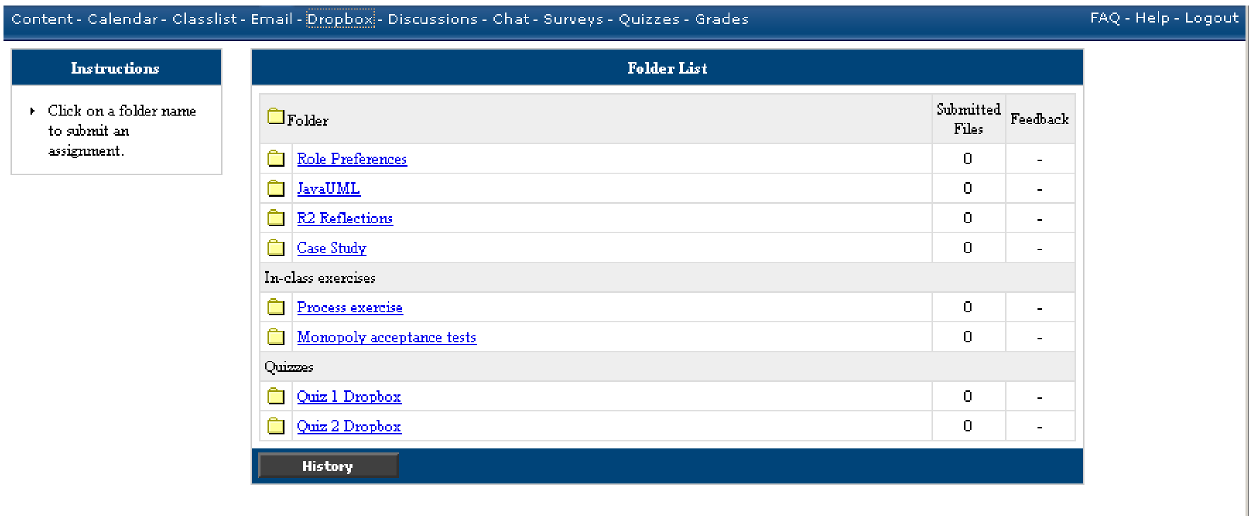

Figure 8. Current organization of student individual work and feedback

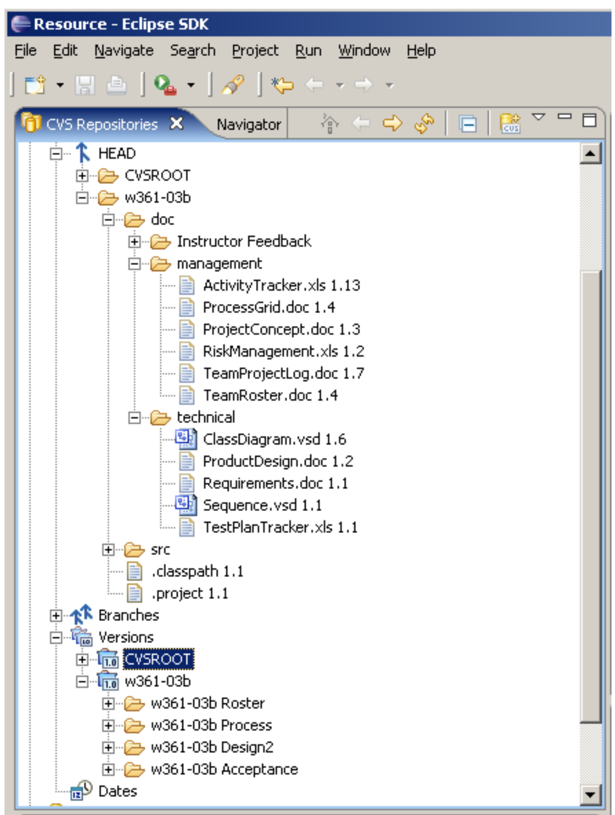

Figure 9. Team project repository structure

In the long run, an implementation of Knowdio could, and perhaps should, be built using a full-featured course management system such as Desire2Learn. Desire2Learn does support learning object repositories, but as a commercial software product, we did not have the ability to add other Knowdio features such as taxonomic search and student-submitted learning content. Hence, we chose an implementation based on DSpace as a quick, effective way to begin to test our concepts in the classroom. If we were to use Desire2Learn or a similar course management system, we would want to provide the students with a view of content other than the fragmented view of a specific course/term, and we would want to provide students a way to add and organize their individual and project content in ways beyond discussion topics and file repositories. We want to integrate content across the curriculum of courses and team projects.

Introduction of Knowdio to the course

As mentioned, we wanted to minimize the impact on the way the course is currently taught, so Knowdio was introduced as a supplement to the current systems, rather than as a replacement. (As we describe later, this resulted in some problems.) The Knowdio principle investigator presented the Knowdio concept and pilot plans in each course section. The ten-minute presentation started with a motivation for using personally-defined taxonomies to tag content, relating the concept to the student's use of labels in Gmail, tags in Thunderbird mail, folksonomies and collaborative or social tagging on del.ico.us, names of directories and files on their personal computer, and Spotlight comments on Mac OSX. We then introduced the concept of using a controlled vocabulary to define tag/subject terms, relating the concept of a controlled vocabulary to the Dewey Decimal System or Library of Congress organization of content in a traditional library, the use of topic navigation to search for content on a news web site, and the use of Wikipedia Categories to browse Wikipedia content. We then asked the (rhetorical) question, "Do you know enough about Software Engineering or Computing to know the best keywords/labels to organize your course and curriculum content?" We then introduced Knowdio using a slide similar to Figure 3, describing the role of SWEBOK to provide an organization of course content. We followed this by a demonstration of Knowdio and a simple exercise to have the students search for a specific item. We then described our plans for the Knowdio pilot, as follows:

- Use Knowdio as a tool to find course content during normal course activities,
- One-third of the way through the course, have a simple survey to see if Knowdio is useful,
- Half-way through the course, use the normal mid-term survey to get another snapshot of Knowdio use,
- Toward the end of the course, have a more detailed survey on the use of Knowdio,
- Optionally, invite students to participate in observation of their use of Knowdio in structured tasks.

In the survey conducted one-third of the way through the course, we asked for responses to three statements with response options ranging from "Strongly Disagree" to "Strongly Agree" on a four-point Likert scale (plus Not Applicable). The statements were:

- I found this tool (Knowdio) to be hard to use.
- It was easy to find the information I needed.
- The SWEBOK taxonomy was helpful in finding the information I needed.

We also provided the students a way to provide open-ended comments.

Survey participation was voluntary. We received 20 responses (out of about 60 students). The survey results are in Figure 10. Here are the open-ended comments from the survey:

- Well, some classes didn't show up. It has the path but nothing inside.
- This tool will be more useful once there is more data.
- Quit using Knowdio.
- Mycourses is good enough, sadly, I don't think that knowdio is really needed. Thanks for the effort though.
- I don't know what SWEBOK is. Knowdio loads VERY slowly, I tend not to use it, there are other, quicker, resources.
- As a student I don't really like going to different places to get the same information. Seeing as how mycourses is already used and it had a nice way to keep in touch with classmates, and already has all the information, I find myself not using knowdio. I think the consensus from the students that I have talked to so far is pick one and it will be fine with us.
- Not all topics are covered. For example I was searching for the difference between case study and user story and it showed no result.
- Looks to be pretty solid, found necessary documents easily enough.

Strongly Disagree	Disagree	Agree	Strongly Agree	Not Applicable	Did Not Answer	
4	8	3	3	2	0	1. I found this tool (Knowdio) to be hard to use
4	5	8	1	2	0	2. It was easy to find the information I needed
2	3	4	3	7	1	3. The SWEBOK taxonomy was helpful in finding the information I needed

Figure 10. Survey results from one-third of the way through the course

The survey responses are quite mixed and somewhat discouraging. After the survey, we continued to monitor student use of Knowdio and found that, by the middle of the course, they had essentially abandoned its use. We decided at that point to suspend any further data gathering, to assess the available results, and plan next steps. We also spoke informally with the course instructors to get their observations.

Assessment of pilot results

There were three primary conclusions we drew from analysis of the available results:

- Providing an alternate way to access course content besides the “system of record” (the existing course management system) resulted in a perceived further fragmentation of course data. Even though the vision of Knowdio is to integrate information silos, the

Knowdio Version 1 pilot created yet another silo. There was no compelling reason for the students or instructors to use the tool.

- A significant number of students (and instructors, surprisingly), did not understand the role of SWEBOK in providing a way to organize and navigate through course content.
- Students expected the system content to be a fully-populated body of software engineering knowledge, rather than the small subset corresponding to the SE-1 course. When they went searching for knowledge (good!) they were disappointed that they did not find what they were looking for (bad!). There is an indication, though, that had the content been more complete, the tool would have been more useful.

Based on these results and the evidence that Knowdio Version 1 was not meeting our objectives of improved learning and more effective teaching, we decided to re-think our deployment and use of the Knowdio testbed.

Try two

In the semester after the initial pilot, the Knowdio principle investigator was scheduled to be an SE-1 instructor. We decided on a number of tasks. We would add more software engineering knowledge (from other courses) to Knowdio, more completely populating the body of knowledge. We would encourage other faculty members to also add their course content. The principle investigator/instructor would use Knowdio in the SE-1 class to demonstrate its potential value and encourage students to use it to answer their own questions, study for exams, and find guidance for performing project tasks. We would encourage students to contribute to Knowdio content and to tag that new content using the SWEBOK taxonomy. Further, the principle investigator would demonstrate and encourage the use of Knowdio in his upper-division and graduate courses and SeniorProjects.

In planning this next step, we knew we were facing a number of new issues. Would other instructors take the time to add course content and tag it using the SWEBOK taxonomy? What compels them to do so? What issues of copyright were we bumping into by making course content available to students who had not taken the corresponding course? (We decided to only share instructor-created content, but even then there were questions about making the information available to students who had not paid tuition for the course or did not have the background to absorb and properly apply information from advanced courses.) What intellectual property protection issues were we facing by students sharing their creations in projects? (At our university, students retain the rights to creative content from their course work and the university and faculty retain rights to course materials that they create. Would students and faculty be preserving their rights when sharing content, and would the system need to enforce tracking of content use?) From a research methodology point of view, having the principle investigator champion the use of Knowdio would certainly make any observations and conclusions suspect. Last, by investing the effort to add and manage more content and open the system up to students to add content, we would need to begin to address issues of access control and version control for dynamically changing content and for protecting unauthorized access to private information (such as grades) or to content that the content author wanted to share only with instructors or teammates. We did not expect to address these issues in the initial pilot, as they would complicate the system and its use before we learned if the overall Knowdio concept was viable.

This last issue, version control of dynamically changing content, hit us early. The course coordinator for the SE-1 course had been working with the instructors to plan changes to the sequence of course topics, some individual assignments, and the content of lectures. In anticipation of that change, as a good engineer, he placed the content in the course web site under version control and then reorganized (refactored) the folder structure. Since the existing course management system used absolute paths in web URLs, and since much of the content in the web site also used absolute path URLs, many of the links in the content held in Knowdio were no longer valid. Further, the content in Knowdio was no longer up to date with respect to the content in the restructured course.

In our initial decision to use a digital library product as the basis of the Knowdio knowledge repository, we knew that it was not a viable long-term approach for dynamically changing content in evolving courses. Digital libraries are designed to favor archival storage of static artifacts, not storage and management of evolving artifacts. That risk manifested itself as the SE-1 course evolved, and we could not stop course evolution for the sake of "scientific research." At this point, we are considering the cost vs. benefit of updating Knowdio with the new content versus providing a more complete content management system in place of the library repository product.

On the other hand, we are now seeing the use of Knowdio by upper-division and graduate students in their project work. They do not need the new SE-1 structure and content (they have already taken the course), and they find value in having other course content available. The items of particular value to them are artifact templates, engineering activity guidelines (such as task lists and review checklists), engineering standards, and example artifacts. Although we have not conducted surveys, anecdotal evidence from discussions and observations indicates that the SWEBOK organization gives them value. For example, to find a template for a Software Requirements Specification or a Risk List, it is very clear to the student where to go in the SWEBOK taxonomy to find that template and to find related content (examples, usage guidance, etc.).

Lessons Learned

Although our pilot use of an initial implementation of Knowdio provided mixed results, we have learned some important lessons to guide our future work. These lessons are captured below.

1. It is difficult to introduce a new student tool into a course, especially if the tool is duplicating a functionally similar tool. If there is not enough clear and compelling reason to change current practice, then the change will likely not occur.
2. In a lesson related to lesson 1 (but from the instructor's perspective instead of the student's), we found that it is difficult to get buy-in and support and participation from instructors who are themselves busy and concerned about providing an effective learning experience for their students. There needs to be a compelling reason for them to consider adoption of an experimental tool. Further, an initial pilot should be limited to one or two course sections led by knowledgeable, motivated instructors if the tool is to be adequately evaluated.

3. A knowledge repository based on archival storage of static content, as with a digital library, is not adequate for the content management tasks necessary for evolving course content and limiting access to those students who have rights to access copyrighted and course-specific content. The repository needs version and configuration control, user access control and administration (probably coupled with the campus registrar for determining what courses a student has taken), and limiting access of student-created content to appropriate teammates and instructors. Further, the repository needs to be integrated with or a core part of the course management "system of record," yet provide a user perspective that crosses the curriculum and provides multiple ways to organize the curriculum content from instructors, students, and external sources. Future implementation of Knowdio on an open source course management system such as Sakai[11] is being considered.
4. "Real" pedagogical research that is assessing the introduction of emerging technology into the classroom requires careful planning and structure, significant commitment from faculty to preserve the structure and champion the effort, and special care to ensure that the student learning experience is not negatively impacted "for the sake of scientific discovery." On the other hand, careful piloting of emerging technology, without the intent of "real" science, has significant value in introducing technological innovation into the classroom.
5. Organizing course content across the curriculum, and using a well-established taxonomy of the discipline as the basis of that organization, seems to make information more readily accessible and understandable in relation to other content.

Conclusions

It is clear that there are needs and opportunities to integrate learning content across the curriculum, including instructor and student contributions. Current solutions fragment the information and complicate the integration of the content into a cohesive whole. Further, the currently available ways to organize content, by course, semester, student, project, instructor, artifact type, etc. are not conducive to providing the student with an integrated mental model of the knowledge of the discipline. Prior research and our Knowdio pilot indicate that using a well-established taxonomy of the discipline body of knowledge may provide a valuable organizational structure for learning content across the curriculum. Being able to tag content with discipline subject headings and search for content by navigating a taxonomy using those terms seems to help students find appropriate content when they don't know the correct terminology. It also seems to encourage students to explore the discipline subjects, resulting in an understanding of the scope, "language," and conceptual structure of the discipline, plus a motivation to browse and discover new learning content.

We believe that our low-cost, quick implementation was appropriate. It provided the same value as iterative software development: early feedback on system functionality and early value. It provided us valuable insight into the requirements of and potential value of Knowdio without a significant investment. Our pilot exposed some of the difficulties with the limited implementation, but it did provide us valuable validation of the key Knowdio concepts. In addition, although the Knowdio pilot did not provide good, hard science on proving the validity of the Knowdio vision, the pilot approach did provide us valuable feedback on the use of this emerging technology and gave us an opportunity to provide value to students without the delays and costs inherent in a more rigorous study. The instrumentation and surveys we used also provide us a testbed that we can evolve and use to conduct good pedagogical research in the future.

Based on the feedback from deploying the initial Knowdio pilot, we are now better prepared for a next step. We clearly need to base the content management on a repository designed for evolving artifacts and with flexible access control capabilities. We also need to add more and more varied learning content to the system, broadening the scope and depth of knowledge available to students.

The feedback from the Knowdio pilot does provide some validation that the approach to tagging content using SWEBOK-based terminology and searching for content using that terminology is solid and appropriate. Given that that was the primary assumption to be validated in our initial experiments, and the assumption seems to be valid in our limited pilot, we can declare our pilot a success, and we are motivated to continue to add features and content to Knowdio and deploy it more broadly across the curriculum. We hope to obtain further feedback on the validity and value of a system that implements the Knowdio vision of marrying content management with a discipline-specific taxonomy to integrate and organize learning content across the curriculum.

Acknowledgements

The authors acknowledge and appreciate the funding and staff support from the Emerging Technology Partnership Program sponsored by Rochester Institute of Technology's Online Learning Department, as well as the Knowdio development effort of Eric Fabricant, a student at the Rochester Institute of Technology, and the effort of the SE-1 (SE361) course students and instructors who evaluated Knowdio in the pilot project.

Bibliography

1. Hawker, J.S., "The Collaborative eNotebook: a Collaborative Learning and Knowledge Management Testbed," Proceedings of the 2006 American Society for Engineering Education Annual Conference and Exposition, Chicago, IL, June 18-21, 2006.
2. A. Forte and M. Guzdial, "Computers for Communication, Not Calculation: Media as a Motivation and Context for Learning," in *Proceedings of the Hawai'i International Conference on System Science*, January 5-8, 2004, Big Island Hawaii, available at http://coweb.cc.gatech.edu:8888/csl/uploads/24/CommNotCalc.pdf, accessed 2006-01-18.
3. Ruiz, F. and Hilera, J.R., "Using Ontologies in Software Engineering and Technology," in Calero, C., Ruiz, F., and Piattini, M. *Ontologies in Software Engineering and Software Technology*, 2006:Springer-Verlag, pp. 62-119. Available at http://alarcos.inf-cr.uclm.es/doc/psgc/doc/lec/parte1/ruiz-springerchap2.pdf (accessed 2008-01-18)
4. Bagert, D.J, Barbacci, M. *et al.*, "Thoughts on Software Engineering Knowledge, and how to Organize it," in *10th International Workshop on Software Technology and Engineering Practice*, IEEE Computer Society, 2002, p. 24. Available at http://www.cs.vu.nl/~hans/publications/y2003/step2002/paper.pdf, accessed 2008-01-18, or at http://doi.ieeecomputersociety.org/10.1109/STEP.2002.1267596.
5. Nilsson, Mikael; Palmér, Matthias; Naeve, Ambjörn; "Semantic Web Meta-data for e-Learning—Some Architectural Guidelines," in Proceedings of the 11th World Wide Web Conference (WWW2002), Hawaii, USA, 2002. Available at http://www2002.org/CDROM/alternate/744/, accessed 2008-01-18.
6. Brase, J. and Nejdl, Wolfgang, "Ontologies and Metadata for eLearning, in Staab, S. and Studer, R. *Handbook on Ontologies*, Springer, 2004, pp. 555-575, available at

http://www.kbs.uni-hannover.de/Arbeiten/Publikationen/2003/Ontologies_for_elearning.pdf, accessed 2008-01-18.
7. See http://www.DSpace.org/
8. Alain, A. and Moore, J.W., Executive Editors, *Guide to the Software Engineering Body of Knowledge*, IEEE Computer Society, 2004 Version, available at http://www.swebok.org/.
9. *Dublin Core Metadata Element Set, Version 1.1: Reference Description* (ISO Standard 15836-2003), available at http://dublincore.org/documents/dces, accessed 2008-01-17. Also, see the Dublin Core Metadata Initiative home page at http://dublincore.org/
10. See http://www.desire2learn.com/.
11. See http://sakaiproject.org/.