

## **2006-1788: THE COLLABORATIVE ENOTEBOOK: A COLLABORATIVE LEARNING AND KNOWLEDGE MANAGEMENT TESTBED**

### **J. Scott Hawker, Rochester Institute of Technology**

Dr. Hawker is an Assistant Professor of Software Engineering at Rochester Institute of Technology. He graduated with a B.S. and M.S. in Electrical Engineering from Texas Tech University in Lubbock, Texas, in 1981 and 1982, respectively. He graduated with a Ph.D. in Electrical Engineering from Lehigh University in Bethlehem, Pennsylvania, in 1990. He has over 15 years of industry experience developing large-scale, multi-agent information and control systems for diverse applications including manufacturing (at Motorola Semiconductor Products and at Honeywell Industrial Automation and Controls), combat pilot decision support and mission management (at Honeywell Defense Avionics Systems), robotics (at AT&T Bell Laboratories), and surveillance (at AT&T Bell Laboratories). 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. In 1999, Dr. Hawker joined the Computer Science Department at the University of Alabama as an Assistant Professor focusing on software engineering, and in 2004 he moved to the Software Engineering Department at RIT. He has worked with the NASA Technical Standards Program applying semantic web, formal modeling, information retrieval, and other advanced information technology to better create, manage, find, deliver, and use standards and lessons learned for aerospace system engineering. He is now applying these technologies in development of tools for collaborative learning in software engineering courses and projects. Dr. Hawker is a member of the IEEE, IEEE Computer Society, ACM, American Society for Engineering Education, Standards Engineering Society, and the Association of Aerospace Standards Users.

# The Collaborative eNotebook: a Collaborative Learning and Knowledge Management Testbed

## Abstract

We envision an eNotebook, a software system that enables students and instructors to manage their learning content across the software engineering curriculum, and to organize the content in multiple ways. We also envision this as a Collaborative eNotebook, which students and instructors use as they collaborate to create, share, and add to this content, and collaborate as they create, share, and add to ways to organize the content. This paper describes the features of a Collaborative eNotebook; it describes a design that integrates existing technologies from digital libraries, advanced search and retrieval, peer-to-peer file sharing, and distributed user identity authentication and access authorization; and it concludes with a description of experiments to assess the effectiveness of the Collaborative eNotebook in knowledge management and learning activities of an introduction to software engineering course.

*“Our knowledge of the world comes from gathering around great things in a complex and interactive community of truth. But good teachers do more than deliver the news from that community to their students. Good teachers replicate the process of knowing by engaging students in the dynamics of the community of truth.”*

Parker J. Palmer, *The Courage to Teach*<sup>1</sup>

## Problem: Capturing, Organizing, and Sharing Information Across the Curriculum

Science and engineering students face the daunting task of incrementally learning and applying a diversity of knowledge about their chosen discipline. In a collaborative learning experience, they must integrate knowledge and learning experiences across courses, within and across project teams, and from external sources such as libraries, professional societies, and standards bodies. In addition, since they are in the process of learning the discipline, they don't yet know the various ways to organize and categorize their growing body of knowledge.

Today's students and instructors use a diversity of technologies to capture, organize, and share information, including computer file systems, course web sites, learning management systems, electronic portfolios, shared project repositories, email, instant messages, blogs, and wikis. Yet, this fragments their information into isolated silos of content with poor, often hierarchical, organization. Typically, each course offering has its own space for content, each project has its own repository, each student and instructor has their own file system, and external sources are only available through web links.

Instead, students and instructors need their learning content to be integrated and organized across multiple aspects of personal and collaborative work, regardless of where the content is stored. A typical student in our ABET-accredited software engineering program must integrate knowledge and experiences from over 30 courses, 15 team projects, and two or more terms of cooperative

work or internship jobs. Students cannot search for and integrate information across the numerous storage and communications silos they currently use for these numerous collaborations. They do not have the mechanisms to catalog, organize, and manage the information, annotate the information with comments and questions, link to external information, or readily share their own information with teammates, instructors, and other collaborators. Finding the information they need, when they need it, is quite difficult. Students and instructors need ways to integrate their learning content across multiple dimensions of knowledge and multiple aspects of collaboration and interaction.

The Collaborative eNotebook is designed to address these needs. It will provide a collaborative learning and knowledge management system that provides powerful ways for students, instructors, and others to manage, organize, and use personal, shared, and external knowledge in their collaborative learning experience. The eNotebook design integrates and extends technologies for digital libraries, advanced search and retrieval, peer-to-peer file sharing, and distributed user identity authentication and access authorization. The Collaborative eNotebook will come pre-populated with valuable learning content from our software engineering undergraduate degree program and from industry standards and best practices. Students and instructors will continuously add to that pre-populated content as they learn and create, together.

The eNotebook content will be organized in multiple ways, reflecting the software engineering course curriculum, the industry consensus captured in software engineering standards (such as ISO and IEEE standards), the IEEE/ACM Computing Curricula models,<sup>2,3</sup> and the Software Engineering Body of Knowledge (SWEBOK).<sup>4</sup> In addition, students and instructors can add to, modify, and extend these pre-built knowledge organizations to provide additional structure and content that reflects the complexity and richness of their knowledge and interactions. The organization will not be a single, hierarchical taxonomic structure of knowledge, rather it will be a multi-dimensional, multi-aspect, multi-purpose collection of multiple ways to organize and access knowledge. Students and instructors will want to, and will need to, access, organize, and contribute to this shared knowledge in collaborative learning and discovery.

With the eNotebook, a student will be able to capture and integrate knowledge across the full curriculum and add to it and share it with project teammates and faculty. They will do a full-text search through their personal and shared eNotebook using discipline-specific and project-specific terminology and context to guide their search and to organize, filter, and interpret search results. Their eNotebook contents will be organized chronologically and by course, as with the traditional paper notebook and courseware or file directories (but spanning much larger time frames than a single bound volume or a single course offering), and their eNotebook will also be organized and sorted by project, collaborator, application domain, theoretical support, development tool used, and any other relevant organizing concept. The students and instructors will easily annotate content with subject identifiers, notes, questions, references, and context, as well as implications, applications, and other data about use and relevance. They will even organize and annotate the annotations.

The ability to easily access and share valuable learning content, along with powerful ways to organize, add to, and manage the learning content, promises to significantly improve the collaborative learning experience. It promises to allow students and teachers to “gather around

great things in a complex and interactive community of truth.”<sup>1</sup> However, this ability may come at a cost: the added level of cognitive engagement and workload to describe, organize, and manage the information, and the potential information overload of having too much information available at once. The eNotebook system will provide features that enable it to be used as a testbed in which to conduct experiments to assess the benefit and cost of the tool in an undergraduate software engineering curriculum. On-going, incremental research will use the testbed to evaluate the effectiveness of the collaborative learning and knowledge management environment and to discover ways to improve the benefit and reduce the cognitive cost. Although the content and organization will focus on software engineering, the underlying tools and technology can be readily applied in virtually any course of study.

Our overall research program, then, has the following objectives:

1. Provide a powerful suite of tools for software engineering students and instructors to collaborate in their efforts to organize, annotate, search, and discover knowledge from their courses, individual learning and research, team collaborations in learning and research, and a broad body of discipline-specific knowledge in standards, best practices, and lessons learned,
2. Provide a testbed of validated knowledge and instrumented knowledge management tools to enable experiments that evaluate the effectiveness of the students and instructors using the collaboration environment in learning and applying software engineering knowledge,
3. Conduct focused experiments to evaluate the effectiveness of these collaborative tools and knowledge objects in helping students meet learning objectives,
4. Based on experiment results and further research, incrementally improve the collaboration tools, knowledge, testbed, and experiments toward creating and assessing innovative and effective collaborative learning materials and tools,
5. Participate in a community of scholars, including faculty and undergraduate and graduate students from multiple universities, collaborating to develop and apply improved collaborative learning materials and tools, working together to disseminate, integrate, apply, and evolve learning materials and tools based on our combined experiences,
6. Accelerate the creation and improvement of high-quality, accredited degree programs and practitioner education in software engineering, which is a discipline that must quickly grow to meet industry needs for highly-productive development of high-quality software.<sup>5</sup>

## **Related Work**

Researchers have long sought to implement Vannevar Bush’s vision from 1945 of a memex, a device to address the “growing mountain of research [where] the investigator is staggered by the findings and conclusions of thousands of other workers—conclusions which he cannot find time to grasp, much less to remember, as they appear.”<sup>6</sup> Bush envisioned the memex as a device which records and stores information and allows the information to be extended, shared, and consulted. Bush described “a memex [as] a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory.” The essential feature of memex, Bush said, is the ability to immediately and automatically tie two information items together into an association. As the user navigates through the information, following existing associations and creating new ones, he leaves a trail of his investigation and the associations made so that he and others can later follow this same trail of evidence and

discovery. The eNotebook project seeks to help fulfill Bush's vision, applied in a university curriculum.

Alan Kay, a computing pioneer (for example, in object-oriented programming, the graphical user interface, Smalltalk, etc. at Xerox PARC) and his colleagues have long-envisioned a machine called a Dynabook: "a portable interactive personal computer, as accessible as a book"<sup>7</sup> focused on learning for "kids of all ages." The current incarnation of that computing and authoring environment is embodied in Squeak, which is called a "media authoring tool."<sup>8</sup>

Mark Guzdial and his colleagues are working to apply the Dynabook vision and Squeak to collaborative learning, providing an environment where students can create and share media across learning groups. They are applying the tools to learning and collaboration in computer science education and have reported significant improvements in learning and retention.<sup>9</sup>

The learning portfolio has long been known to provide a valuable way to collect a student's work across the curriculum so that they and their instructors can reflect on and assess the student's learning and provide concrete artifacts that reflect that learning. Electronic portfolios (ePortfolios) are electronic versions of traditional hard-copy portfolios. Upchurch and Sims-Knight report on the use of ePortfolios in a software engineering course, but the scope of content and reflection was limited to only one course.<sup>10</sup> Reflection across the curriculum was not discussed, and apparently was not supported by their ePortfolio software. McDonald has evaluated the use of hard-copy learning portfolios in assessment across the software engineering curriculum.<sup>11</sup> He identified difficulties in the assessment process, some of which could be solved by having an automatic electronic record in a curriculum-wide ePortfolio, such as providing multiple ways to organize and navigate contents, and being able to recover missing artifacts.

The Kepler project<sup>12</sup> is studying the use of digital libraries for individuals and small communities, bridging the gap to digital libraries for large organizations (universities, companies, etc.). Kepler enables users to self-archive content and provide a federated access to content published by a group of collaborators. The Kepler vision has influenced the eNotebook vision, and we plan to re-use some of its open-source implementation in our implementation.

Early work on electronic engineering notebooks, such as the SHARE project at Stanford,<sup>13</sup> showed the value of electronic capture and sharing of information in collaborative product development. The Design Space Colonization project at Stanford is now going beyond this to provide a "DesignJournal" which integrates the electronic notebook with computer-based engineering design tools.<sup>14</sup> Electronic notebooks are generating significant interest and are beginning to be adopted in industry, as evidenced by a number of commercial products, the series of Electronic Lab Notebook Conferences, and the formation of the Collaborative Electronic Notebook Systems Association.

In our work on providing software development process information to software engineers,<sup>15,16</sup> we found that simply providing linked information content in a web site is often not effective. To address this, we defined and exploited an underlying model of content meaning, purpose, and organization, and of engineering task workflow, to more effectively organize and deliver process information to software engineers. In our work for NASA on DynaLL,<sup>17</sup> we are exploring the use

of technologies to allow flexible organization and capture of engineering lesson learned content, combined with consistent integration, search, and access to lessons learned from across NASA. In our work for NASA on SA\_MetaMatch,<sup>18</sup> we exploit common meaning of terms in document metadata and document indexes to automatically discover documents that are likely to address related topics. This experience gives us confidence in the value of an eNotebook, and it gives us experience with the technologies to implement eNotebook.

The rest of this paper describes our approach to the design of an eNotebook testbed solution and our approach to using the testbed to evaluate the effectiveness of collaboration and learning using eNotebook.

## **eNotebook Design Approach**

In building the eNotebook software system, we will adapt and integrate existing functionality into an integrated solution. All the parts are available in some form. We will incrementally integrate them into an evolving knowledge management and collaboration environment; populate the environment with valuable learning content; conduct controlled experiments observing students, instructors, and others collaborating in a new learning experience that the environment enables; and analyze experiment results to guide us toward improved collaborative learning tools and techniques for the next increments of development and experimentation.

We have an overall design structure that identifies a number of existing technologies that we are integrating into the overall solution. We have delivered and are beginning to use an early increment of the eNotebook and are using it as a shared content repository within our project team. The following figures illustrate our approach to developing the Collaborative eNotebook software system. Figure 1 shows that collaborators interact directly with each other and indirectly through the eNotebook content and content description and organization. This content interaction and organization is built upon a data distribution and access control infrastructure.

Figure 2 illustrates the architectural layers of the eNotebook design and how collaboration between users is facilitated by integration and interaction at each layer of separate installations of the eNotebook software. The bottom layer provides for storage of knowledge content and metacontent (content descriptions and organizations). The second layer provides for content and metacontent distribution and access control. The third layer provides for content search and subscription. The top layer provides the tools for content and metacontent presentation and creation.

Figure 3 shows the software packages that we chose to implement each of the layers in Figure 2. Each package provides specific functionality and is available from basic operating system services (such as file systems and TCP/IP communications) or from open-source components (identified in the diagram and discussed in a later section).

We certainly don't intend to detail and implement this design all at once. Rather, we are building a series of thin vertical slices across the layers, providing incremental capability and a testbed for incremental experience and improvement in collaborative learning using eNotebook capabilities.

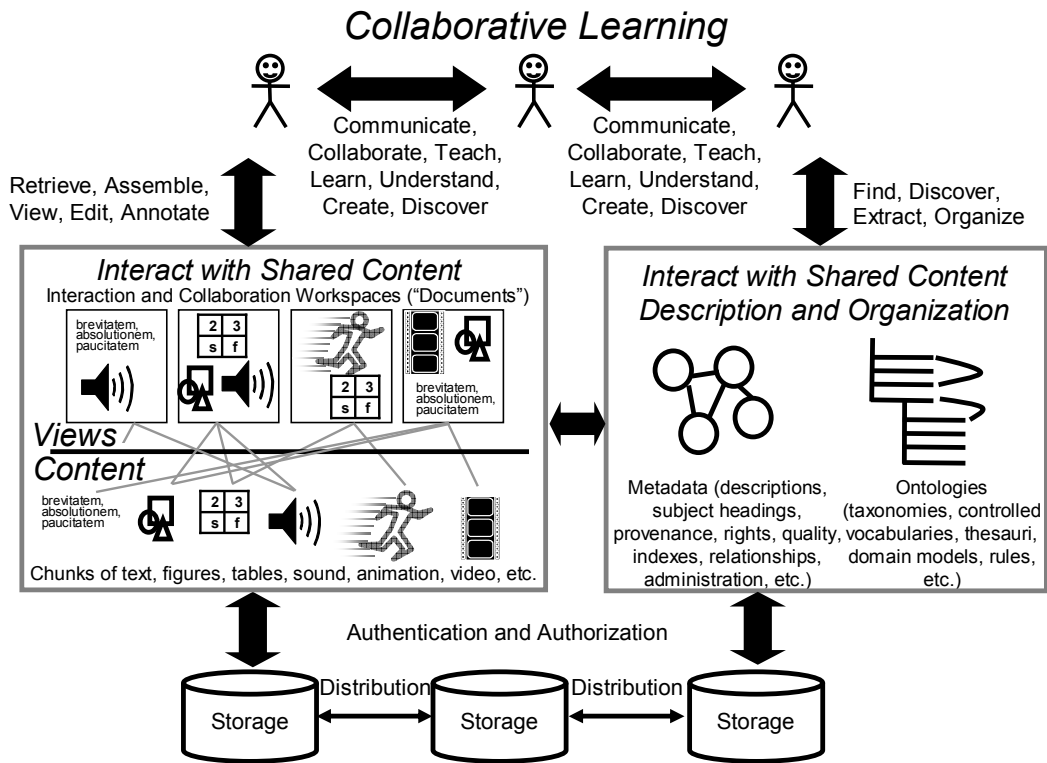


Figure 1. Improve collaborative learning by enabling interaction with shared content and with shared description and organization of content.

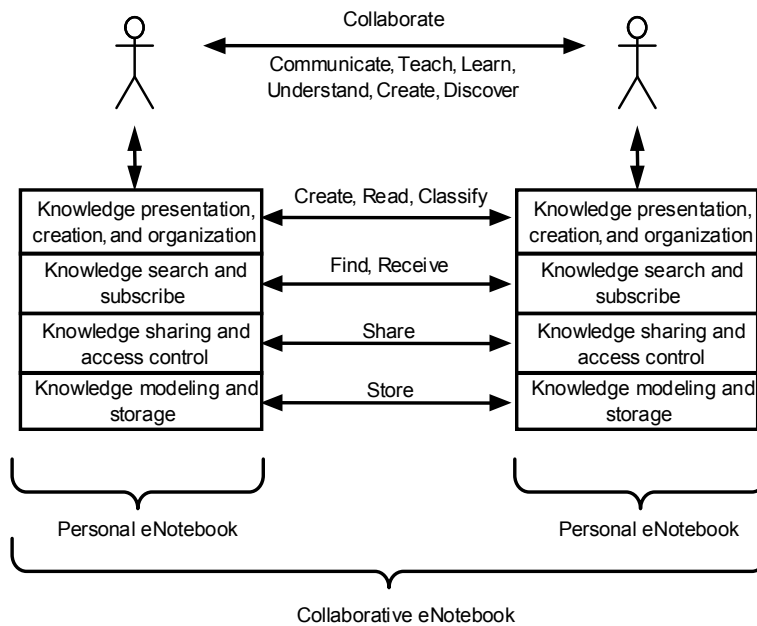


Figure 2. Architecture layers in eNotebook

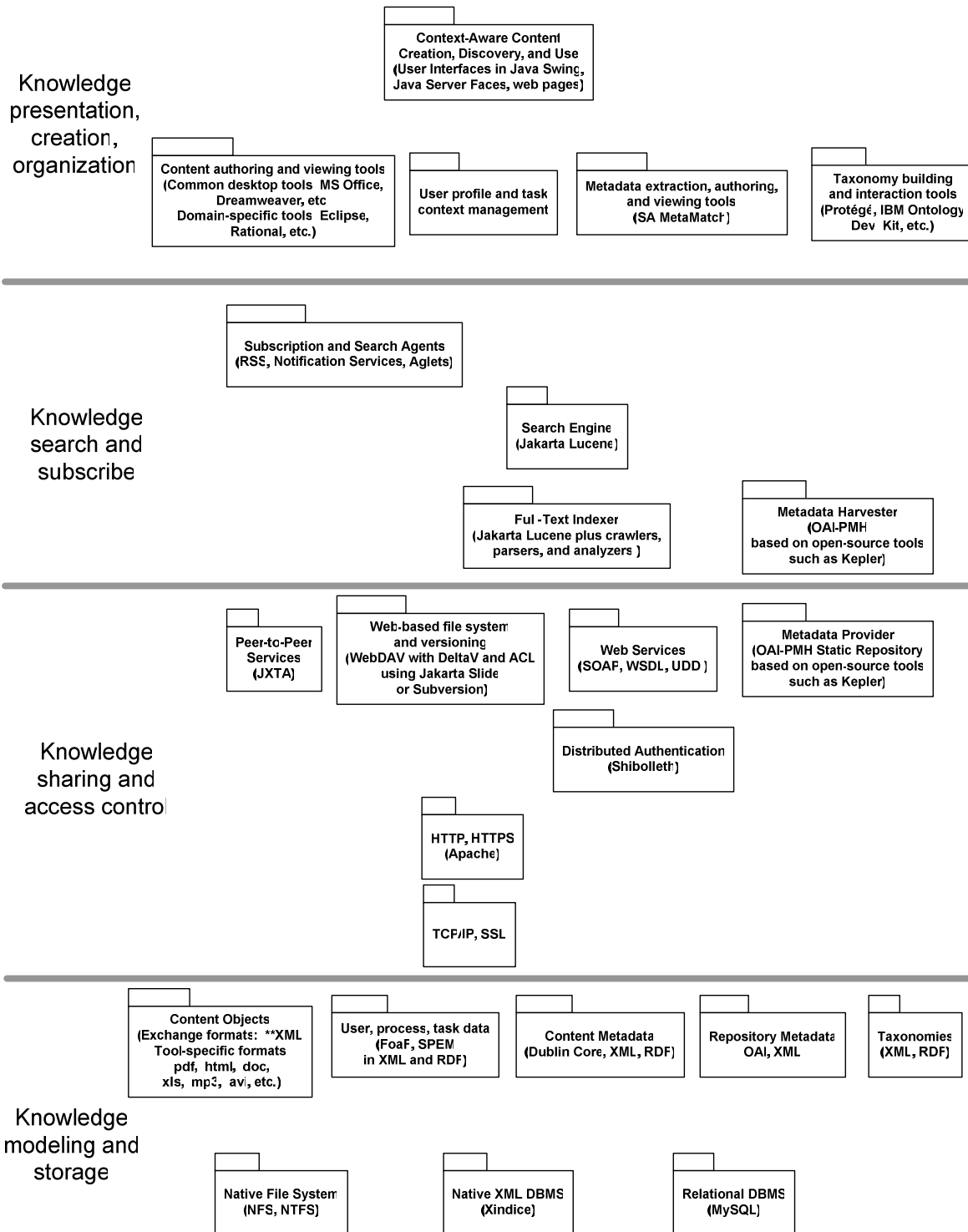


Figure 3. The functional elements in the architecture layers and existing implementations (in parentheses) that we will adapt and integrate



## eNotebook Materials

The materials in a student's eNotebook are traditional course materials (lecture notes, readings, personal notes, graded exams and assignments, etc.) and project team artifacts (shared project deliverables, documentation, activity plans, instructor feedback, etc.). The materials also include shared materials such as team discussions and chat session archives, links to external materials (on the public internet and on protected intranets, such as items available only through the university library or through the department and university's protected and licensed systems), and common materials across the curriculum and the discipline (course listings, lessons learned and assessment repositories, best practices, industry standards, etc.).

In particular, an early increment of the eNotebook content will provide common content for our Introduction to Software Engineering course, including course syllabi, learning objectives, lecture notes, in-class activities, reading materials, and learning review quizzes. It will also include common team development process artifacts, such as activity guidelines, templates, checklists, and sample artifacts. This content will be available across all course offerings and beyond the academic term for the course. When access control features are built and verified in eNotebook, we will also provide access to licensed and copyrighted content, including textbook lecture materials, materials from the IBM Academic Alliance, etc. We will also provide links to external materials including the IEEE Software Engineering Standards, SWEBOK, SWEnet course materials, and the North Carolina State University's Software Engineering Open Seminar Project materials.<sup>20</sup> We will also test tightly-integrated search and access to external document servers including the NASA Technical Report Server, Open Access Journals, and our campus Digital Media Library. Note that all these materials will be indexed as part of the overall eNotebook content organization, even if the actual materials are external and under controlled access. Of course, students and instructors can add content to their personal eNotebook content repositories and share that with their collaborators as they choose.

A key feature of the Collaborative eNotebook, and a critical element of its power, is that it provides more than "raw" content. It also provides "metacontent" including content metadata (such as title, creator, summary, subject, access rights, etc.), relationships between content objects, indexes of document terms for use in search and discovery of meaning, and statements of the origin and quality of content. It also provides ontologies—taxonomies, shared vocabularies, glossaries, and models specific to the software engineering domain—of the content. This metacontent provides a powerful, flexible way for students and instructors to organize and use learning content. For example, by selecting terms from a standard software engineering glossary to define content subjects, content can be categorized and classified with other similar content, and team members get assurance that they are using the same terms and meanings in classifying their content and in searching for content. By using the terms and content taxonomies from industry-consensus standards and documents, such as, IEEE standards, SWEBOK, and the Computing Curricula, they are working within a common organizational context for the discipline while they are still learning about that discipline. They can then use these terms to classify, organize, search, and use personal and shared information.

In particular, we will incrementally develop multiple taxonomies of software engineering terms based on SWEBOK,<sup>4</sup> the Software Engineering Curriculum Guidelines,<sup>2</sup> the IEEE Standard

Glossary for Software Engineering Terminology,<sup>21</sup> and the course descriptions in our curriculum. All built-in content will be organized using these taxonomy terms as subject headings, and eNotebook users can use the same terms (and terms they and their collaborators have added) to organize their added content.

## **eNotebook Tools**

The students and instructors using the eNotebook also will share the tools they use to inspect, shape, and manage their learning materials. The eNotebook provides access to the tools to search, navigate, organize, create, inter-relate, and disseminate the team's collaboration knowledge and course knowledge. The tools to create and view materials include discipline-specific design and analysis tools, such as computer-aided software engineering tools, and traditional office automation tools such as word processors, drawing tools, and email clients. Tools to organize and inter-relate content include tools to describe the content (to build the metadata), build search indexes, categorize the content (label its subject using shared and private vocabularies), discover and declare related content (references, versions, similar subject matter, etc.), and create associations between related content. This also includes tools to build and manage the domain ontology elements including taxonomies, controlled vocabularies, and models. Tools to search and navigate content include keyword-driven search engines and taxonomic (classification-oriented) browsing, search and relevance filtering based on personal profiles and task context, automatic search agents that transparently discover relevant content, visual clustering and connections of related content, and navigation across content references. Tools to disseminate content include content description (metadata) publication and harvesting, peer-to-peer file sharing with group authorization policies, and traditional web-based and shared file access.

The taxonomy will be built using open-source software tools such as Protégé.<sup>22</sup> Our goal will be to provide a relatively flat (lightly structured) vocabulary of shared terms (as opposed to a linguist's highly structured, formal, rigid taxonomy) so we can experiment with the balance between formality and usability of users interacting with taxonomies and using them to guide content organization and discovery.

We will also provide mechanisms to define personal taxonomies, and multiple shared taxonomies. The taxonomies and taxonomy building and browsing tools will be instrumented so we can study the creation, use, evolution, and sharing of taxonomic elements, as well as their use in collaboration tasks.

We are defining a learning content metadata model based on the Dublin Core Metadata Initiative elements,<sup>23</sup> with additional metadata elements derived from the IEEE Standard for Learning Object Metadata,<sup>24</sup> and the US Department of Defense's SCORM (Sharable Content Object Reference Model) metadata model.<sup>25</sup> The metadata will be especially valuable for organizing content by assigning Dublin Core "Subject" terms from the software engineering discipline taxonomy described above. This metadata model will allow us to experiment with users interacting with metadata content that was explicitly created to define its utility in learning. We can see how metadata is used to summarize content, place it in a learning context, and use it as a

reference to more complete or related content. It will also give us the basis to evaluate how extensive and rich the metadata needs to be to provide benefit in discovery and understanding.

We will provide metadata, annotation, and relationship capturing tools that allow us to measure the amount of effort content creators and readers spend to capture metacontent. The metadata capture tools will be derived from the MetaGen and AutoGen tools we built for the NASA Technical Standards Program<sup>18</sup> and from open source software available for building Dublin Core-compliant metadata.<sup>23</sup> Annotea will be the base of the annotation tool.<sup>26</sup> We will adapt the Dynamic Lessons Learned system<sup>17</sup> that our collaborators at the University of Alabama are developing for NASA, which allows a lesson author the flexibility to select content elements and element types to include in a lesson or to define new element types that meet their needs, yet which also provides consistent structure and organization that allows a lesson user to find lessons relevant to their interests and tasks. In the eNotebook testbed, we will be able to identify which lesson elements are most often used by authors and which are most relevant to users.

We will provide an information sharing infrastructure and tools for students, faculty, and others to maintain and allow controlled access to personal and shared learning content and metadata content. The content sharing infrastructure and tools will be based on open-source implementations of the WebDAV (Web Distributed Authoring and Versioning) family of standards<sup>27</sup> such as Jakarta Slide<sup>28</sup> or Subversion<sup>29</sup>, open-source implementations of the evolving JXTA standards for Java-based peer-to-peer file sharing,<sup>30</sup> secure user authentication and distributed authorization services for universities based on Shibboleth<sup>31</sup> from the Internet2 efforts, (which are part of the National Science Foundation Middleware Initiative for shared cyberinfrastructure<sup>32</sup>), and peer sharing groups defined using the Friend-of-a-Friend (FoaF) definitions of person, group, project, and organization.<sup>33</sup> By explicitly specifying who has access to what content, both for building search indexes and for document access and annotation, we will be able to experiment with different levels of user control of access to personal and shared artifacts.

Our search tools are based on open-source indexing, crawling, and search engines, including tools from the Apache Lucene<sup>34</sup> effort and SWISH-e.<sup>35</sup> We create separate indexes of documents, of metadata elements (for example, the keywords in a title, separate from the keywords in an abstract), and give relevance weight to terms and phrases that appear in the domain taxonomies and user interest and task profiles. We will also leverage this information to cluster possibly related documents. These capabilities will build on our NASA SA\_MetaMatch document discovery tools.<sup>18</sup> This will also give us the details and control to experiment with fine-grained user interaction in specifying and evaluating the results of search and browse behaviors.

We are now implementing metadata provider and harvesting functions based on open-source implementations of the Open Archive Initiative (OAI) Protocol for Metadata Harvesting (PMH)<sup>36</sup> emphasizing lightweight, personal library implementations such as Kepler.<sup>12</sup> We will build a static PMH repository provider, and we will experiment with various configurations of personal, locally shared, and department-wide metadata repositories. This will allow us to design experiments to understand the user's need for metadata-based summaries versus access to the full document to evaluate document utility for their task, as well as experiments to

understand the user's need to understand the partitioning of their shared document space into personal, team, course, department, and other peer groups.

We will provide a mechanism for users to define interest profiles that help to search and filter information. The student will define their interests and current tasks by selecting terms from their personal and shared taxonomies, by recording reusable search queries, and by defining preferred scope of search (specific projects or courses, for example, or specific repositories or external web sites). This information on a student's interests and tasks can also drive automatic search and discovery agents that work in the background to find information for a student and present it to them in their context of use. This "publication" of students' interests and tasks allows information to be "pushed" to the student, which complements the students explicitly searching for ("pulling") information.

### **Evaluating the Use of eNotebook**

The rich characterization of knowledge, rich knowledge relationships, and explicit declaration of collaboration partners in eNotebook enables a rich experimentation testbed for collaborative learning and knowledge management. We must ask the question, though, will it improve peer-to-peer learning in our undergraduate software engineering curriculum? Critical to our testing of this hypothesis is our prediction that realizing these improvements in learning will require that students and instructors spend additional effort to describe, organize, and share information, but that that effort is not excessive, and it pays for itself in improved information organization, search, and discovery. We will build into the eNotebook data gathering instrumentation to provide an experimental testbed to investigate the Collaborative eNotebook's impact on learning and to explore ways to use the eNotebook that improve learning and reduce the additional effort.

Our experiments will focus on assessing the effectiveness of students in collaborative learning, in general, and, in particular, on assessing changes in effectiveness when using eNotebook to share and manage information in collaborative learning compared to not using eNotebook. Here, we describe our general approach and some specific areas for experimentation.

Through surveys, interviews, focus group discussions, observation using contextual inquiry,<sup>37</sup> and data gathered from our existing learning management system and course web sites, we will build a baseline characterization of tools and materials students use in team projects and collaborative learning without the use of eNotebook. We will also use surveys, interviews, focus groups, observation, and data gathered from eNotebook and related tools (learning management systems and course web sites) to characterize collaborative learning with eNotebook.

Some of the specific items we will assess follow: Who shares what information, why, and when? What are the motivations and restrictions to sharing? Are there issues of information ownership and pride (or embarrassment) in sharing? Who will use the system the most, and what can be done to encourage use by others? What usage guidelines and policies will encourage eNotebook use? We also need to assess the quality of the implementation of eNotebook: Are the correct, useful materials and tools available? Are they usable? Is the system stable, secure, reliable, etc.?

Since we are particularly concerned with measuring the cognitive work associated with creating and using content metadata and organizations, we will focus on experiments that compare the effort of “on-the-fly” search and application of relatively unorganized information versus the upstream cost and downstream benefit of organizing information for later use. We will formulate experiments, surveys, and interviews to identify the sources of problems and success, including focus on the learning materials; visualization and content of metacontent (descriptions, relationships, annotations, etc.); tools for content and metacontent visualization, manipulation, and navigation; the underlying software engineering concepts being learned; team coordination and communication topics; etc.

Here are some of the team exercises for our Introduction to Software Engineering course which we will investigate:

- Use on-line instructions and lessons learned in setting up and testing the shared software development environment including a Java integrated development environment (Eclipse) and a software version management system (CVS),
- Use taxonomy-guided search of information and lessons saved by prior students to select a software design tool for team design,
- Use process guidance and shared authoring and annotations to collaborate to define team member roles,
- Use process guidance, a shared activity planner/tracker, and a shared status report template to record team efforts, to develop a team weekly status report, and to collaborate in updating weekly project plans,
- Use shared team tools to package software project artifacts (requirements documents, programs, test plans, etc.) into a test release, then make that release available to another team for testing, capture test results, receive test results back, and engage in discussions among the development team and test team to understand test results and to plan how to address identified software defects.

Data we will gather to obtain insight on collaboration and knowledge creation and management will include:

- The amount of information in personal and shared eNotebook spaces and the rate of growth of that information
- Characterization of what information is captured and linked to in relationships
- The amount of information referenced versus copied/replicated
- Resource hit counters (web page clicks, views, and downloads, for example)
- Team software development productivity metrics and other traditional software engineering metrics
- User requests for help in using the system
- User change requests, feature requests, and problem reports for the system
- Traditional usability measures for human-computer interaction

### **Development Plans and Status**

Given our concern that we must be cautious in our introduction of new cognitive workload in our collaboration tools, and given the very large scope of our planned development effort, we are using an incremental development and experimentation process. Our incremental process also

reflects the cyclic model for knowledge production and improvement of practice in education.<sup>38</sup> Our focus is on creating new learning tools that make existing materials available in novel ways, and then assessing the impact of those tools and materials on collaborative learning.

We are incrementally developing testbed functionality, evaluating its effectiveness, and then, based on experimental results, we will incrementally add or modify testbed functionality and continue experimentation. We have implemented our first increment, which provides a shared document repository with a metadata tagging facility and full-text and metadata-driven search. The metadata implements the simple Dublin Core metadata element set. We are using this repository within the eNotebook research project to capture relevant literature, product artifacts (requirements (developed as part of a Software Requirements course project), architecture design, object class and interaction design, code baselines, etc.), meeting minutes, and email threads. We now are implementing an OAI-PMH metadata harvester/provider pair and plan to deploy our next product release to our collaborators at the University of Alabama so we can test shared, distributed repository search and access. We are also now building an initial software engineering taxonomy based on the SWEBOK models, along with a content annotation tool to tag content with subject terms from the taxonomy. We will then instrument these tools and design experiments to assess eNotebook effectiveness within our research effort. We will then enhance the content and begin to roll the product out for experimental use in select courses, while we continue our development, adding in distributed repositories, access control, and other features in later increments.

## Conclusions

We have defined an ambitious project to develop a Collaborative eNotebook, which we expect to be a powerful environment for knowledge management and collaborative learning. The broad vision and success of related efforts, such as Squeak (Dynabook), ePortfolios, electronic lab notebooks, personal and group digital libraries, and learning management systems, give us confidence that combining these capabilities into a coherent environment, and populating this environment with valuable, well-organized learning content, will be an important contribution to software engineering education. We have identified readily available technology elements and learning content which we can integrate into an evolving eNotebook implementation. We have defined an approach to experimentation using the evolving implementation so that we can measure and improve on the anticipated benefits of eNotebook. We are eager to have the tool, obtain results, and use them to the benefit of the software engineering education community.

## References

1. Parker J. Palmer, *The Courage to Teach: Exploring the Inner Landscape of a Teacher's Life*, San Francisco: Jossey-Bass Publishers, 1998, p. 115.
2. Joint Task Force on Computing Curricula, *Software Engineering 2004, Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*, August 23, 2004, a volume of the Computing Curricula Series from the Association for Computing Machinery (ACM), and the IEEE Computer Society (IEEE-CS), available at <http://sites.computer.org/ccse/> accessed 2006-02-24.

3. The Joint Task Force for Computing Curricula 2005, *Computing Curricula 2005, The Overview Report including The Guide to Undergraduate Degree Programs in Computing*, a cooperative project from the Association for Computing (ACM), The Association for Information Systems (AIS), and the IEEE Computer Society (IEEE-CS), September 30, 2005 available at [http://www.acm.org/education/curric\\_vols/CC2005\\_Final\\_Report2.pdf](http://www.acm.org/education/curric_vols/CC2005_Final_Report2.pdf), accessed 2006-02-24.
4. A. Alain and J.W. Moore, Executive Editors, *Guide to the Software Engineering Body of Knowledge*, IEEE Computer Society, 2004, available at <http://www.swebok.org/>, accessed 2006-02-24.
5. Daniel E. Hecker, "Occupational Employment Projections to 2014", *Monthly Labor Review* (Bureau of Labor Statistics), November 2005, available at <http://www.bls.gov/opub/mlr/2005/11/art5full.pdf>, accessed 2006-02-24.
6. Vannevar Bush, "As We May Think," *Atlantic Monthly*, July, 1945 available at <http://www.theatlantic.com/doc/194507/bush>, accessed 2006-01-18.
7. Alan Kay, as quoted in "The Dynabook Revisited—A Conversation with Alan Kay," in *The Book and The Computer*, available at <http://www.honco.net/os/kay.html>, accessed 2006-01-18.
8. For more on Squeak, the web site <http://squeak.org/> is for computing experts, and the site <http://www.squeakland.org/> is for children, parents, and teachers.
9. 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.
10. R.L. Upchurch and J.E. Sims-Knight, "Portfolio Use in Software Engineering Education: An Experience Report," in *Proceedings of the 32<sup>nd</sup> ASEE/IEEE Frontiers in Education Conference*, Boston, MA, November 6-9, 2002, available at <http://fie.engrng.pitt.edu/fie2002/papers/1464.pdf>, accessed 2006-01-18.
11. J. McDonald, "Use of Student Portfolios for Outcomes Assessment of a Software Engineering Program," in *Proceedings of the 2005 American Society for Engineering Education Annual Conference and Exposition*, June 12-15, Portland, Oregon.
12. K. Maly, *et al.*, "Kepler – A Communal Digital Library," *Proceedings of European Conference on Research and Advanced Technology for Digital Libraries (ECDL) 2004*, Bath, UK, pp. 317-328, available at <http://www.cs.odu.edu/~mln/pubs/ecdl04kepler.pdf>, accessed 2006-01-17. Also, see the Kepler web site at <http://kepler.cs.odu.edu/>.
13. SHARE: A Methodology and Environment for Collaborative Product Development, available at <http://www-cdr.stanford.edu/SHARE/share.html>, accessed 2006-01-18.
14. See the DesignJournal web site at <http://www-cdr.stanford.edu/DesignJournal/>.
15. Hawker, J.S., Ma, H., and Smith, R.K., "Web-Based Process and Process Models to Find and Deliver Information to Improve the Quality of Flight Software," (Best Paper in Track) 2003 Digital Avionics Systems Conference, October, 2003.
16. J.Scott Hawker, "Integrating Process, Product, and People Models to Improve Software Engineering Capability," Computer Science Technical Report TR-2002-05, University of Alabama, September, 2002.
17. Swapna Gupta, J. Scott Hawker, and Randy K. Smith, "Acquiring and Delivering Lessons Learned for NASA Scientists and Engineers: A Dynamic Approach," *Proceedings of the 2005 ACM Southeast Conference*, Kennesaw, GA, March, 2005.
18. Christina (H.Y.) Yau, and J. Scott Hawker, "SA\_MetaMatch: Document Discovery through Document Metadata and Indexing," *Proceedings of the 2004 ACM Southeast Conference*, Huntsville, AL, March, 2004.
19. Gregory Hislop, *et al.*, "Work in Progress – Developing the Community of Software Engineering Educators," 35<sup>th</sup> *Proceedings of the ASEE/IEEE Frontiers in Education Conference*, October 29-22, 2005, Indianapolis, IN, pp. T3G-24-TG3-25, available at <http://fie.engrng.pitt.edu/fie2005/papers/1421.pdf>, accessed 2006-01-17. (SWEnet is the Network Community for Software Engineering Education: <http://www.swenet.org/>)
20. M. Rappa, *et al.*, "OpenSeminar: A Web-Based Collaboration Tool for Open Educational Resources," *Proceedings of the First International Conference on Collaborative Computing*, San Jose, CA, December 19-21, 2005, available at <http://open.ncsu.edu/mrappa/collaboratecom2005.pdf>, accessed 2006-01-17. (The OpenSeminar project at the North Carolina State University is developing course materials for Software Engineering and other disciplines. See <http://openseminar.org/se/> for the software engineering content.)
21. *IEEE Standard Glossary of Software Engineering Terminology* (IEEE Std 610.12-1990).
22. See the Protégé home page at <http://protege.stanford.edu/> and see the series of *Protégé Conference Proceedings* such as at <http://protege.stanford.edu/conference/2005/index.html>.

23. *Dublin Core Metadata Element Set, Version 1.1: Reference Description* (ISO Standard 15836-2003), available at <http://dublincore.org/documents/dces>, accessed 2006-01-17. Also, see the Dublin Core Metadata Initiative home page at <http://dublincore.org/>, and see the series of *Proceedings of the International Conference on Dublin Core and Metadata for e-Communities* available at <http://dcpapers.lib.sfu.ca/>
24. *IEEE Standard for Learning Object Metadata* (IEEE Std 1484.12.1-2002).
25. *The Advanced Distributed Learning Sharable Content Object Reference Model*, 2004 version, and associated standards available at <http://www.adlnet.org/scorm/index.cfm>, accessed 2006-01-17.
26. M.R. Koivunen, "Annotea and Semantic Web Supported Collaboration," *2<sup>nd</sup> European Semantic Web Conference 2005, Workshop on User Aspects of the Semantic Web*, Heraklion, Greece, May 29, 2005, available at [http://www.annotea.org/eswc2005/01\\_koivunen\\_final.pdf](http://www.annotea.org/eswc2005/01_koivunen_final.pdf), accessed 2006-01-17. See also the Annotea home page at <http://www.w3.org/2001/Annotea/>.
27. HTTP Extensions for Distributed Authoring – WebDAV, IETF RFC 2518, The Internet Society, <http://www.ietf.org/rfc/rfc2518.txt>. See also IETF RFC 3253 — Versioning Extensions to WebDAV (<http://www.ietf.org/rfc/rfc3253.txt>) and IETF RFC 3744 — WebDAV Access Control Protocol (<http://www.ietf.org/rfc/rfc3744.txt>).
28. See <http://jakarta.apache.org/slide/>.
29. See <http://subversion.tigris.org/>.
30. JXTA v2.0 Protocols Specification, available at <http://spec.jxta.org/nonav/v1.0/docbook/JXTAProtocols.html>. See also the JXTA web site at <http://jxta.org/>
31. See the Shibboleth web site at <http://shibboleth.internet2.edu/>
32. See the NSF Middleware Initiative home page at <http://www.nsf-middleware.org/>.
33. FOAF Vocabulary Specification, July 2005, available at <http://xmlns.com/foaf/0.1/>, accessed 2005-01-17. See the FOAF web site at <http://www.foaf-project.org/>.
34. See <http://lucene.apache.org/>.
35. See <http://www.swish-e.org/>.
36. Open Archives Initiative (OAI) Protocol for Metadata Harvesting (PMH); see <http://www.openarchives.org/>.
37. H. Beyer and K. Holtzblatt, *Contextual Design: Defining Customer-Centered Systems*, Morgan Kaufmann, 1997.
38. Loewenberg Ball, D. Chair of the RAND Mathematics Study Panel, *Mathematical Proficiency for All Students: Toward a Strategic Research and Development Program in Mathematics Education*, RAND Corporation, 2003.