# Distributed Collaborative Homeworks

## Learning Activity Management and Technology Support

John M. Carroll (IEEE Fellow), Marcela Borge, Craig Ganoe, and Hao Jiang
Center for Human-Computer Interaction and College of Information Sciences and Technology
The Pennsylvania State University
University Park, Pennsylvania 16802 USA
jmcarroll@psu.edu

*Abstract*— **We describe use of partially distributed collaborative assignments in a Usability Engineering course with respect to (1) learning activity management and (2) technology support.**

***Keywords- collaborative learning, case-based learning, collaborative software, learning technoloy, usability engineering***

### I. INTRODUCTION

The engineering students of today will most likely work in partially distributed teams during their careers, as network mediated collaboration becomes more routine. During the past three years, we have been experimenting with partially distributed team assignments in an upper-level undergraduate course in usability engineering at the Pennsylvania State University (Penn State). These assignments employ *case-based learning* [Carroll & Rosson, 2005; Jiang et al., in press]: students analyze and apply ideas from online case studies documenting real-life professional practices in usability engineering to their own design projects. The assignments also employ *distributed collaborative learning* [Ganoe et al., 2003; Xiao et al., 2008]; the students are asked to work together outside of class using collaborative software over a period of time ranging from several days to several weeks to develop a design analysis or prototype, and a report describing their work.

An example assignment we have used asks student teams to develop a user interface prototyping strategy for a web-based information system. Each team member was asked to review a different case study in which prototyping was employed in order to identify candidate prototyping ideas and approaches, and propose these to the team. The team was then to evaluate, select, and integrate these proposals and compose a joint report.

In this paper, we describe our experiences with case-based distributed collaborative homework assignments in the context of our problem-based usability engineering course. We also describe our experiences developing and employing an online library of case resources and collaborative technology to support these assignments and the other activities in the usability engineering course. We conclude with a summary of lessons learned to date, and some of our plans for upcoming versions of the course.

This research was supported by the National Science Foundation of the United States under awards 0231111 and 0354195 from the Division of Undergraduate Education (DUE), and under award 0735440 from the Advanced Learning Technologies (ALT) program, and by the Edward M. Frymoyer Chair Endowment. During the writing of this paper Carroll was Shaw Visiting Professor in the Department of Information Systems of the National University of Singapore.

### II. LEARNING ACTIVITIES

We are addressing three types of learning objectives. First, we want students to learn about and practice applying specific usability engineering concepts and techniques (like user interface prototyping). Second, we want students to exercise and develop their collaborative abilities (for example, learn to critically evaluate their team's ideas). Third, we want students to get experience working in teams using distributed collaborative software applications.

We have found that the activities we give students to do need a lot of scaffolding to get them to think more deeply about core course concepts (scenarios, design tradeoffs, etc.). Students, feeling they already know more than they actually do, are more focused on completing the assignment that understanding the underlying concepts of the activity. With respect to usability engineering, this is a well-known challenge: students think that because they are people, they will be able to understand and address the needs of users.

Students also overestimate their knowledge and skills with respect to their collaborative abilities and collaborative software. In general, our students have too simple a view of how to effectively collaborate. Thus, if collaborative process-goals are not explicitly stated, students may simply divide up the work and engage in minimal interactions. Experiencing these interactions are an important objective since they will need to understand users and engage in peer reviews during their careers.

Finally, students are highly experienced with social-networking systems, but rarely have they tried to coordinate activity through such systems. They know how to communicate and interact online, but not necessarily how to coordinate and carryout intellectual collaborative endeavors. Usability engineering rarely takes place in a closed room. Potential users, developers, designers, etc. can be globally distributed, and all play roles in the process.

Pushing students to recognize their knowledge gaps requires reflective activities, modeling, and explicit prompting. Motivating them to develop their abilities is a more difficult task, but holding them accountable for all three learning objectives is a crucial component to providing an authentic usability engineering experience.

### A. *Case-based Approach to Usability Engineering*

Our usability engineering course was designed to be case-based. We created a collection of online case descriptions and made these available to the students. The cases were originally presented via an ASP (Active Server Pages) based Usability Case Studies (UCS) Library website (http://ucs.ist.psu.edu/). There are seven cases, though one is quite incomplete. We choose a variety of domains – mobile banking, telephony (PhoneWriter), online community (TappedIn), online business (Garden.com), nonprofit animal rescue (PAWS), as well as an online version of the virtual science fair prototype case which is also described in the students' textbook [Rosson & Carroll, 2002].

Each of the case studies is a separate hypertext organized, like the path the course follows and like the student semester project requires, by phases of the usability development process. These phases include requirements analysis, activity design, information design, interactions design, documentation, and user testing. Each phase describes how a particular design team developed its scenarios, worked with users, and ultimately validated its design through empirical testing with actual users. For further details on the design of the case studies see [Rosson et al., 2004; Carroll & Rosson, 2005].

Many of the homeworks and in-class activities in our usability-engineering course make use of the case library materials. For example, in the 2009 course, the second homework asked students to study the Tapped-In case and identify at least two conceptual metaphors the designers used to clarify and envision approaches to their requirements. The students were asked to explain how the metaphors contributed to this phase of the design activity. Another homework asked students to identify interaction design issues in the mobile banking case study, and to analyze the underlying tradeoffs for each issue. Students were asked to describe how each issue was resolved, or not resolved, in the actual design work.

Case materials help learners to engage with domain content and to vicariously experience domain practices and situated concepts through a narrative [Carroll & Rosson, 2006]. These cases also serve as models for students of how core concepts can be applied in practice. We found this approach to be attractive to students and effective with respect to learning outcomes [Carroll & Rosson, 2005]. One qualification is that our students have a somewhat narrow conception of what is up-to-date. Thus they sometimes perceive the cases as being less relevant to them since they describe system development projects and practices from the early 2000s. Though a valid criticism, this viewpoint also reflects students' lack of experience and difficulty in generalizing concepts.

### B. *Project-based Approach to Usability Engineering*

Our usability engineering course originated in 1995 at Virginia Tech. Our design concept was to organize the course following the information and activity flow of a scenario-based system development process. We used scenarios as an integrating representational technique, since they emphasize the user's point of view and the importance of describing and supporting work activities as the user sees and experiences them [Carroll, 1995].

Our course starts with user requirements, and scenario-based methods to investigate and describe the users' mental model. The course then progresses to upstream design where user activities are designed by envisioning scenarios, but specific technology, display, information layout, and user interaction decisions are not considered. Following this stage of pure activity design, the scenarios are elaborated with user-interface details. Subsequently, design scenarios are implemented as prototypes, and evaluated with respect to a wide range of usability issues and criteria.

The course was organized around a semester-long system development project. As the course works its way through a curriculum organized by the flows of system development process, student teams define, implement, and test their projects. Roughly, a third of the way through the course they hand in a requirements plan describing how they worked with users to develop their design concept. Two-thirds of the way through the course, they hand in a design document describing how they developed their concept into a prototype. Finally, at the end of the course, students hand in an evaluation document describing how they tested their prototype with users. This course has been successful with computer and information science students. In 2002, we published a textbook based on the course [Rosson & Carroll, 2002]. For details on the course plan see http://ist413.ist.psu.edu/.

During the past two years, the course has changed in two significant ways. First, we have further integrated homework assignments and in-class activities with the semester project. The students take the semester project quite seriously and the grading is weighted to encourage them to do this. However, this makes the homeworks and in-class activities seem like distractions rather than smaller activities intended to help them develop concepts and skills required for the project work. Thus, we reorganized the content of many of the homeworks and in-class activities to more directly support the project work, hoping that this would improve students' engagement with these activities.

The second change we made to the course was to include out-of-class collaboration beyond the semester project. The semester project has always involved out-of-class student collaboration. However, in the 2009 version of the course we included three homeworks that were to be carried out collaboratively over a two-week period. These homeworks were designed to support the semester project-planning that the students were doing, but they were also separately graded deliverables from the project.

### C. *Taking Collaboration Seriously*

When we first started using a collaborative project-based approach to the Usability Engineering course in 1995, we provided general coaching to student groups as to how they should try to collaborate. We suggested they should share all of their ideas, listen to others' views, criticize constructively, try to be sure that everyone did his/her part, and make their group work greater than the sum of the individual parts. In general, the student teams did behave cooperatively, and appeared to enjoy working together both in class and outside class. They frequently became passionate about their projects, and clearly spent much time and effort on this part of the course.

However, there also were recurring problems. Some students contributed little or nothing, and allowed their team members to do all the "joint" work. The remedy for this problem is to weight individual grades according to a student's team contribution, which can be assessed by querying each team member about the other members' contributions. Our students are uneasy with being asked to assess their peers, but this can be mitigated by providing the rationale for this assessment. A complementary approach is for the instructor to observe of group work sessions and try to note who is actively engaged in the joint work. Even just noting attendance at group work sessions can be a meaningful proxy for assessing contribution.

A more subtle problem was that most teams used a management strategy of "divide and conquer". The key joint work they did was to divide the project up into nearly independent subprojects, each of which could be carried out by a single student (i.e., non-collaboratively). At the conclusion of their work they more or less stapled the pieces together and handed the ensemble in as their "collaborative" project.

In the 2006 version of the usability engineering course we designed three distributed collaborative homeworks. Our objective was to provide the students with an opportunity to practice collaboration, and specifically to encourage them to try to use collaborative software. The collaborative homeworks involved analyzing case materials in the online case study library, and creating a collaboratively-authored document using an open source toolkit that supports integrated synchronous and asynchronous interactions BRIDGE (Basic Resources for Integrated Distributed Group Environments) [Ganoe et al., 2003].

There were three collaborative face-to-face activities, each of which had the same basic organization. Each group met in class to plan an approach that would be carried out during the following five days. On the fifth day, the group met again in class to finalize and hand in its results. Each of the activities involved analyzing a case study and trying to apply lessons learned to the team's semester project, and each involved some role-playing. For example, one activity requires members to review a case from the perspective of one of the following roles in a development project: product manager, usability manager, documentation manager, customer/user. Adopting their role's perspective, students were asked to focus on how key issues of documentation design were handled in the case, annotate the case documents, and download them to a collaborative environment.

The aim was for students to see the power of asynchronous collaborative work through such activities. We wanted them to be engaged by the roles and through the roles to experience the inherent conflicts and tradeoffs in decision making in the development process. These activities also provided them with opportunities to function as a thoughtful professional, reflecting on lessons learned in past projects and creatively extending and applying those lessons to new project contexts. In our assignments, the case studies provide a surrogate for prior usability engineering experience that the students generally do not yet have.

These activities did not work as planned. Most of the student teams simply did not carry out the distributed collaboration, but instead tried to cram their work into the two face-to-face meetings we had scheduled for them to coordinate their work. A major factor in this was that we did not explicitly weight collaboration in the grade for this activity. We expected that the opportunity to use collaborative software and engage in collaborative writing would be intrinsically attractive enough to the students that they would at least try these activities. Thus, a lesson we took away from this was that it is important to explicitly assess and grade any collaborative activity that is essential to a course component: providing students with concrete rubrics and expectations for performance beyond the product. This means emphasizing that how they execute their activity as a process is just as important as whether and when they get it done.

### *D. Distributed Collaborative Homeworks*

In the Spring 2009 version of the Usability Engineering course we introduced three new distributed-collaborative homeworks. These were different from the 2006 activities in several respects. They involved far more complex analysis of the case studies, and therefore could not possibly be carried out in class. Each homework was intended to take between two and three weeks to accomplish. Also, we explicitly told the students that we would inspect their collaborative environments and assign grades based on their collaborative processes as well as their outcomes or products. We reclassified these assignments as special homeworks, instead of activities, which called attention to their status as out-of-class work. Reclassifying them as homeworks also made them worth more points toward the final course grade.

Students had to conduct a complex analysis of the case studies in the case studies library. They had to review different phases of design across different cases identifying ideas and techniques in the case studies that could be applied to their team's semester project. Even though the case studies were developed to address other stakeholders and other requirements, the cases could still be utilized as a model of how general course concepts would be applied in real world settings. For this reason we suggested they use the cases creatively, ensuring that their analysis also be informed by their own team's requirements analysis and field work with users and other stakeholders.

In order to ensure equity in participation we tried to structure that activity such that all members needed to make specific contributions. For example, each team member had to recommend user interface design ideas and design rationales to the team based on their individual analysis. They were told to first gather and consider all the different ideas that members had identified, and then prioritize, select and adapt ideas and techniques to their own project. The final result of this particular homework was to envision a user interface design with at least two user interaction scenarios.

This set of collaborative assignments worked better than the 2006 assignments: Every group actually carried out the assignment and produced a collaborative workspace populated with documents. However, the assignment did not work as planned. They were intended to take about three hours, distributed across three weeks. We used a jigsaw design [Slavin, 1980] in which each student carried out distinct but essential individual investigatory activities, the results of which had to be pooled and integrated to reach the specified result. The students were told that we would assess evidence of their collaborative process in the document versions and chat logs in their BRIDGE workspaces.

Their deliverables demonstrated that the teams gathered information, but there was more scattered evidence of synthesis and critical evaluation of the individually gathered information. One out of eight teams clearly integrated concepts from the case studies library with their semester project. Most teams handed in either a summary aggregation of the individual information that was gathered, or a planning statement for how the information would be applied in the design work on the group's semester project. Furthermore students spent little apparent effort identifying which ideas from their individual analyses were the best ideas to carry forward into the group project. Most groups tended to satisfice, to adopt the first reasonable idea identified.

### *E. Collaboration as a First Class Course Topic*

We have found that one of the challenges in making collaboration a course outcome is that many students believe they know how to collaborate. Their understanding of collaboration, however, is "divide and conquer": they see the key to working together as a matter of breaking down the overall project into parts, and are clearly assigning responsibilities that are as independent as possible. They are confident that they can do group work in this fashion, and are therefore impatient about being coached and provided with cognitive scaffolds to enable better collaboration. This is actually a formidable teaching challenge: The students think they have a basic professional skill that they really do not.

In the 2009 version of the course we addressed this directly by introducing an articulated model of collaboration [Borge & White, 2009; Carroll et al., 2008]. We analyze effective collaboration as consisting of effective communication (team member's build on each other's ideas and work to develop a joint understanding ), planning (the group's activity is directed by an agenda of goals), productivity (the team stays on track with respect to task goals and ensures work quality), and evaluation-negotiation (different perspectives among members are made visible and addressed, and the group's results are critically assessed). These four facets of collaboration can be embodied in actual roles that students adopt [Borge & White, 2009], in our usability- engineering course we have treated them more abstractly as four essential qualities of the collaborative interaction.

In the 2009 version of the course, we made a systematic effort to teach the four facets of collaboration. In week 2 of the course, student teams videotaped their own interaction as they worked on an in-class activity on requirements change (they were presented with one further requirement for the garden.com design, and asked to analyze the impact of this requirement on the upstream design. In week 3, they were given a brief lecture on the four facets of collaboration and video-based training on the facets (they viewed student team interactions from a prior version of the usability engineering course to see the collaborative facets modeled and then to classify snippets of team behavior.

In the third week of the course, student teams were given their own team interaction videos to review with respect to how they enacted the four facets of collaboration. We felt that a direct self-confrontation would help the students to recognize that their own collaborative skills could indeed be improved. This collaboration thread of instruction during weeks 2-4 of the course was actually fairly lightweight. The lecture we provided was only a few minutes long, and the other activities were interleaved with discussions and other activities focused on various other usability engineering topics.

In week 7 of the course the student teams started the first of the three distributed collaborative homework assignments, as described in II-D above. The instructions for each of these homeworks asked the students to make clear in the writing they did in their collaborative workspaces how they were using the four facets of collaboration in carrying out the homework activity. They were asked to make their collaborative process clear in the chats and other work products they created.

## III. TECHNOLOGY SUPPORT

Through the past three years, we implemented our distributed collaborative homeworks using an open source toolkit that supports integrated synchronous and asynchronous interactions (Basic Resources for Integrated Distributed Group Environments, BRIDGE, [Ganoe et al., 2003]). The software supports distributed collaborative team authoring of reports as well as commenting on and construction of new usability cases. Over the history of our usability engineering course, we have explored the integration of these collaborative tools for team work with the content of the UCS Library website. We have found that collaborative software is sometimes too tightly integrated to be effectively used by student teams. Email in particular is problematic in that students rely on it, but already have email clients and accounts they use outside the collaborative suite. We are currently developing an open, web-based collaborative workspace to help students integrate their existing tools and practices with support for collaborative learning.

### *A. Read-only Usability Case Study (UCS) Library*

The usability case study library (http://ucs.ist.psu.edu/) was originally implemented as a ready-only repository using Microsoft's Active Server Pages (ASP) technology. We wanted to exploit the flexibility of hypermedia to allow students to study the cases at various levels of depth, drilling down on demand. This is depicted in Figure 1.

In the figure, a student is investigating the garden.com case study. The student has navigated to "envisionment work" in the information design phase of the case study, using the indented list widget on the extreme left of the window. This displays a

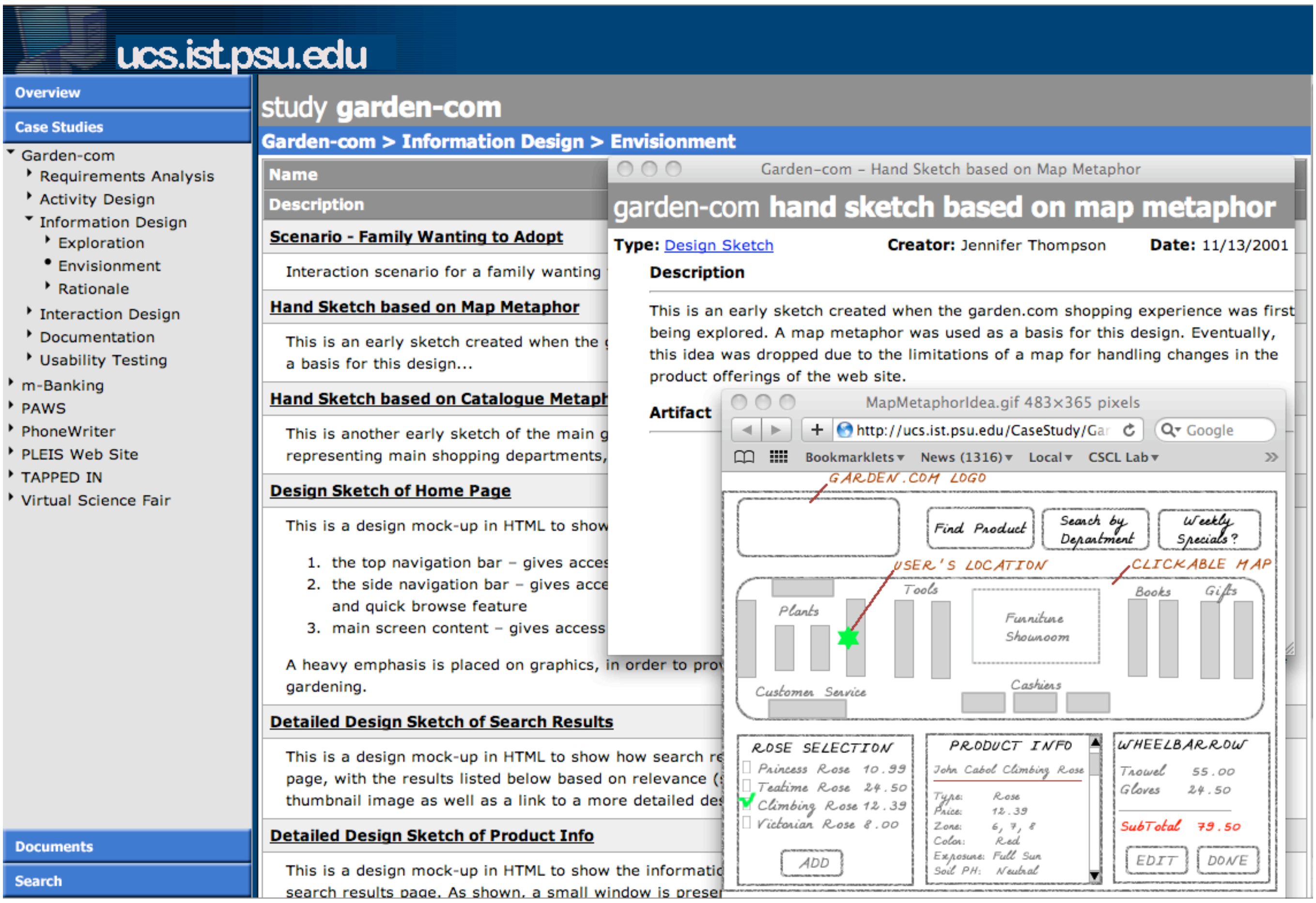


Figure 1. The original, read-only, Usabilty Case Studies (UCS) Library.

series of design artifacts, documents, sketches, notes, and so forth that were produced in the garden.com project during the envisionment part of its information design phase (viewable in Figure 1 in the large right-hand pane of the UCS window). The student is studying a particular design artifact, a hand sketch made by one of the designers illustrating the map metaphor that the designers used during this phase of the design. The description of the hand sketch appears in a pop-up window that displayed on top of the main UCS window. The student has opened the hand sketch, which displays in a pop-up on top of both other open windows.

We have had success with project-based and collaborative case-based assignments using this UCS library [Carroll & Rosson, 2005]. However, through using it we also identified several areas for further development. The original does not support several learning activities that we believed would be useful, and that are highly consistent with the active learning approach we have adopted. Students cannot annotate case-study objects, thus they cannot make notes in the browser as they read case study materials, they cannot highlight or mark them in any way for future reference (including sharing their reactions with fellow group members), and they cannot easily return to materials they have examined before.

Most importantly, students could not create their own cases. Once we started using case-based approaches it seemed obvious to consider having students create a case study analysis of their own semester project. Starting with the 2005 offering of the usability engineering course, we have had the student teams create case study reports of their own projects. Students have reported that this activity helps them to reflect on their own work. However, the original UCS library neither supported multiple user accounts nor convenient means for project teams to create and edit cases. During 2005 and 2006, we jury-rigged functionality to allow students to create cases. As students could not create cases, teachers and practitioners from entities other than Penn State could not either. We know from informally asking colleagues and from a survey we conducted in 2005 that many instructors do use cases in teaching usability engineering (and human-computer interaction). We wanted to make it easier for others to contribute content to the UCS library.

### B. *Collaborative Case Builder*

For the 2006 offering of the usability engineering course, we created a collaborative case builder using the BRIDGE toolkit (see III-D for detailed on BRIDGE). This was a partial solution to some of the design challenges of the ASP browser.

The collaborative case building allowed students to download case study content from the UCS into their own BRIDGE workspace where they could collaboratively annotate it, and even directly edit the source content. This functionality allowed student groups to carry out the distributed collaborative activities described in II-C above.

However, a major limitation of this approach was that the case study objects edited in BRIDGE could not be uploaded. Thus, students could work on an editable version of a case study, and could collaborative create their own case study materials as well. But they could not directly save or render this work to the UCS Library website. That step still had to be carried out through a manual, jury-rigged process. Also, potential case authors from outside our university still could add content to the case library on their own.

### C. *Editable Usability Case Study (UCS) Library*

During 2007-2008 we developed an editable UCS Library [Jiang et al., in press], addressing the limitations of the original UCS. The new system focuses on allowing users (students and/or educators) to create cases and supporting more authentic learning activities (more usability engineering related practice, more social interaction, more reflection, etc). There are four aspects we especially take into consideration in our design: social interaction, authentic activities in usability engineering, resource accumulation and updates, and communities of practice. For a detailed discussion, see [Jiang et al., in press]. These four perspectives are tightly related and contribute to each other. The functional categories in Table 1 summarize the main functions and design concerns in our UCS redesign, which are affordances for our educational and learning goals.

TABLE I. NEW UCS LIBRARY FUNCATIONS AND REQUIREMENTS

| Category | Brief Description | Requirements (Touchstones) |
|---|---|---|
| Case schema | Capturing case study activities, procedures and key contents (documents, scenarios, artifacts, etc) | Authenticity |
| Distributed case authoring | Providing means to allow users to contribute remotely | Resource accumulation and updating |
| Comments and tagging (Shared) | Providing channels for users to communicate and share information | Community of Practice, Social interaction |
| Administrative functions | Providing administrative and security functions (authentication and authorization, version control, etc) | Supportive functions |

*1) Case schema and conceptual framework:* In order to capture and support the authenticity of usability-engineering practice, the software follows the phases of the usability process described in II-A. Rosson and Carroll (2002) formulated this case schema for usability engineering case studies, in which the key stages and activities of usability engineering are integrated together into a case structure. This case schema captures the flow of usability-engineering activities and types of documentation corresponding to them. The case schema is represented in the newly designed UCS library similar to the one in the read-only UCS library described in III-A. Students can navigate throughout a case with this schema on the left of the window shown in Figure 2. The team assignments usually cover all stages and activities described in the case schema. For example, when the student team starts to work on their projects, it begins with requirement analysis phase and the students work on detailed activities such as gathering information from user site, interviewing intended users, etc. The structured case schema provides scaffolding for the students to follow the usability engineering processes.

*2) Online case authoring:* Online case authoring allows students, and other users who have proper access level, to remotely contribute case material to the library through standard web browsers. Our read-only UCS did not provide a convenient authoring method to the wider usability engineering community. This inherently limited the source of usability cases. The addition of proper access controls increases the opportunity for remote contribution of cases to the library. Now, case authors from different institutions and educational sites can contribute to the library.

Users with proper authorization can create new case materials, edit existing content, and delete content objects. During usability engineering activities (interviewing, on-site note taking, videotaping, etc), multimedia data may be collected and uploaded to the system and then referred to with URL links. In order to fully capture and support authentic materials, the UCS system supports digital objects in different media formats, such as video, audio, rich text formats, etc.

The system provides a web user interface for students and instructors to author cases. Thus all the actions needed to create a case are available through a standard web browser. For example, to modify an existing document in a case, someone who has the access can just click edit button on the page which turns the document into editing mode; after editing, one can just click the save button to save it. The UCS Library server will automatically keep a retrievable version as well.

Allowing online case authoring not only brings more case resources, but also provides a means to increase the sense of community [McMillan 1996; Rovai 2002]. Allowing users to create contents is a step toward full participation and engagement within the community, because in our situation, creating a case covers all activities we described in the case schema. In the UCS, we also capture information such as the contributor of content, which gives case authors visibility and credit.

In our usability engineering course, student teams' semester long projects are usually usability engineering projects from real-world. Thus, they essentially create new cases. During the project, students work on each phase and produce reports on their projects. After two semesters, student groups have created 15 cases in the library.

*3) Commenting and Tagging:* The system allows users to contribute and communicate by commenting on and tagging of

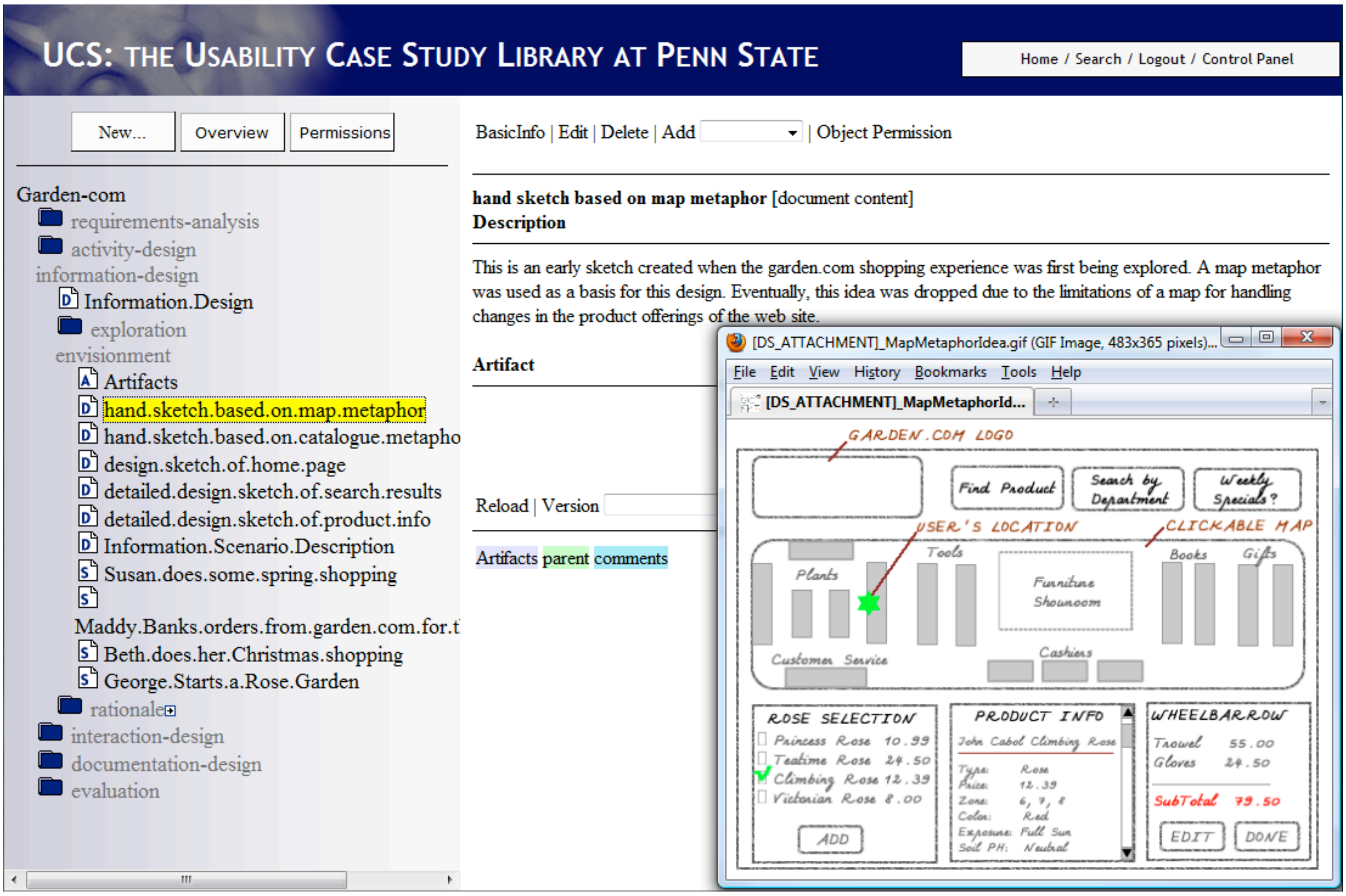


Figure 2. Updated UCS Library with user editing, and commenting features.

case contents. Users who have access to a case can make comments on it, providing their thoughts, ideas, suggestions, criticisms, and so on. Other users can review these comments and add their own. Since comments can be shared across users, the comments can promote discussion and convey information during collaboration.

Tags can help users by providing coded reminders to content or as a way to organize information of interest. In our design, the system will support both public and authorized tags. An example of an authorized tag might be where an instructor would want to set up tags that will only be used by students in her class to find the relevant content for a lesson. Tags will be stored as metadata of the objects. Some services related to tagging is searching on tags, filtering content by tags and displaying all the valid tags in a user's workspace or on the main page for a case. In this way, comments and tags improve the case studies by extending the memory of a user and enabling the user to write down their thoughts, ideas and other reactions on the cases.

*4) Administrative functions:* Besides functionalities we mentioned above, the system also needed other administrative functions, to make the whole system work smoothly. Those functions are, for example, version control which records histories of all digital objects and from which users can retrieve those histories; authentication and authorization subsystem that secure case contents, etc.

With these basic functions implemented, we deployed the system on an Internet-accessible server. The newly deployed system has been in service for two semesters, spring 2008 and spring 2009, to support usability engineering education. The size of our courses is about 45 to 50. Each week, the instructor selects materials corresponding to the topic of that week from the case collection and asks students read them and reflect on them. Along with other activities and readings, students are asked to review existing cases hosted in the UCS Library and reflect on those cases. In their distributed homework assignments, members of student teams are asked to reflect on case materials and pool ideas contributing to their ongoing projects. For example, in the week we introduce usability evaluation, the students go through the evaluation section in different cases (e.g., Garden.com, Tapped In, Phone Writer, PAWS, m-Banking, Virtual Science Fair), and they are asked to comment on the approaches used in those cases and come up with ideas and approaches of their own when they are in those situations.

The student team semester projects are applications coming from real-world contexts: commercial companies, non-profit organizations, etc, and they are already put into use or on

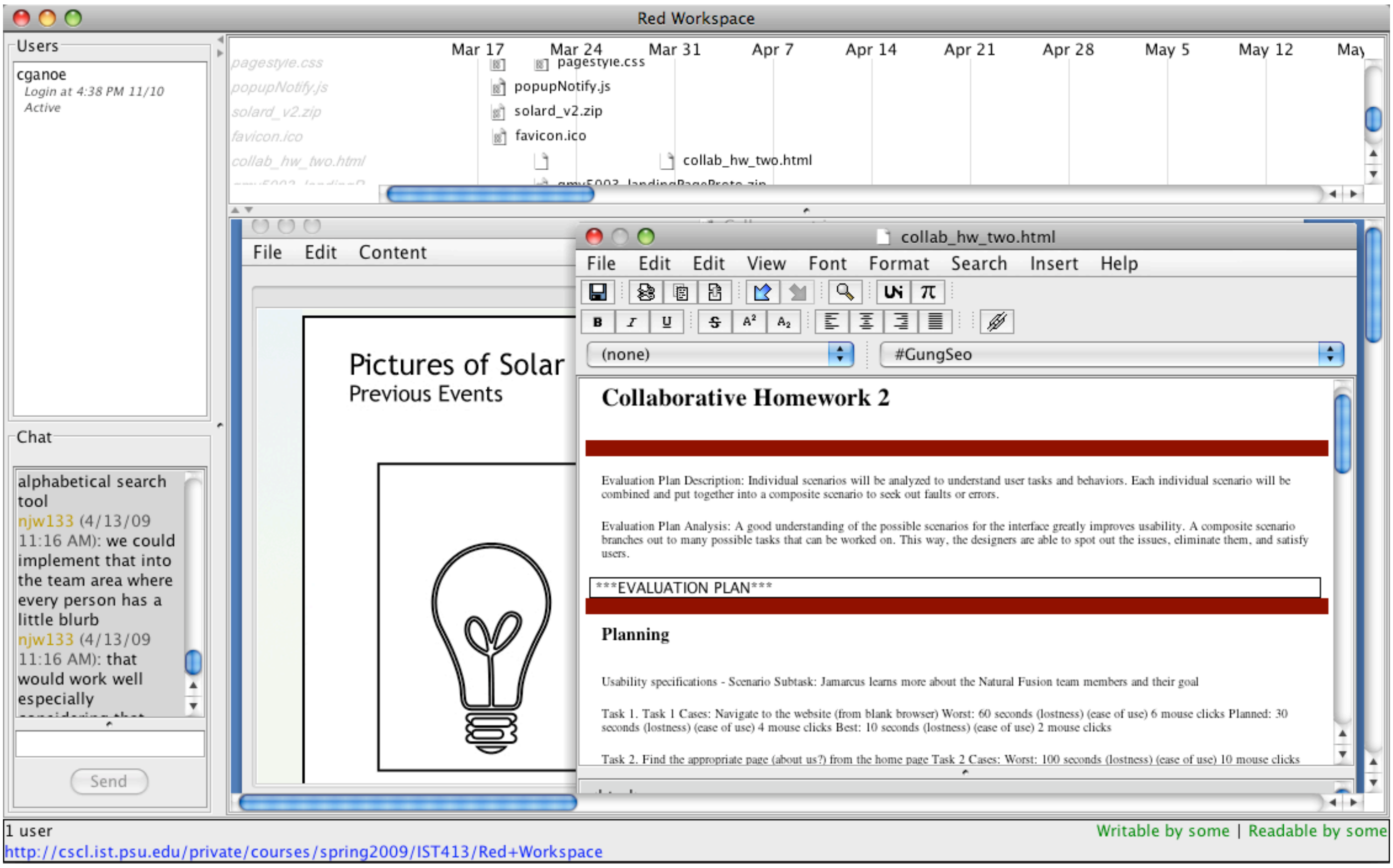


Figure 3. Students develop an evaluation plan in the BRIDGE workspace for one of the collaborative homeworks.

testing. We asked students to work with these organizations and discover possible usability improvements, design/redesign them and evaluate their new designs. The students post their semester-long projects into the UCS Library, so those projects become valuable case resources. In spring 2008, the seven student teams contributed seven cases to the UCS library. In 2009 spring, eight cases were contributed to the library.

### *D. BRIDGE*

In the 2009 version of the usability engineering course, the three distributed collaborative homeworks, described in II-D, were carried out using BRIDGE (Basic Resources for Integrated Distributed Group Environments) [Ganoe et al., 2003], an open source system that supports the kinds of collaborative activities that we are trying to encourage and support among the usability engineering students (Figure 3). Our design for these learning activities requires members of the student teams to be able to individually contribute to the group work. For example, a team might develop a document analyzing how PAWS website designers carried out their activity design, information design and interaction design phases of the non-profit's project. Team member contributions might be made at any time during the 2-3 weeks of the assignment, and their contributions need to be immediately accessible and editable by their fellow team members.

In addition, we wanted the team members to have access to lightweight interaction tools, such as persistent text chat, so that they could exchange ideas, raise issues, and comment on work that was planned or completed. These interactions need to be shared and persist both synchronously, if they happened to be in the workspace at the same time, and asynchronously, if they happened not to be working at the same time.

Finally, we want students to be able to develop the team's final product collaboratively, both synchronously and asynchronously, as possible. In all three of the distributed collaborative homeworks, teams were to create a final document, synthesizing the work that each member had carried out and directing this analysis towards the team's semester project. As mentioned above, we also want the teams to tell us how they had used the four collaborative facets in organizing their work. We left it to the teams to decide exactly how to do this; for example, it could have been done in the chat and/or embedded in the final homework document.

## IV. FURTHER WORK

We are currently planning for the Spring 2010 version of our usability engineering course. Our planning is directed at refining our approaches to both learning activities and learning technology. With respect to learning activities, we want to more explicitly articulate the presentation of the collaborative homeworks to help students experience a more sophisticated collaborative interaction. We also want to integrate the collaborative homework activities even more closely with the semester project. With respect to learning technology, we want to create a lightweight, browser-based shared workspace for distributed collaboration.

### A. *Plans for learning activity design*

One development vector in the usability engineering course over the past two years has been greater integration of homework assignments and in-class activities with the group semester project, as described in II-A. We want to further pursue this. First, the students have a huge amount of work to do in this course, and integrating that work under the rubric of the semester project makes it more coherent for them. Second, the semester project is intrinsically motivating to the students; by ensuring that homeworks and in-class activities are more tightly bound with the semester project we may be able to leverage this to increase students' overall motivation.

In the 2009 version of the course, the distributed homeworks were separate assignments that pushed students to apply concepts important to the project, but not directly connected to project deliverables. We want to rectify this in the next iteration. As part of their final semester project, students have to create a case study of their design process; our plan is to design the distributed collaborative homeworks so that they directly contribute to the students' case studies. The distributed homeworks would be used as opportunities for students to reflect and evaluate on their own projects. Students would be expected to incorporate the language of the course to explain their design processes and apply important concepts like trade-off analyses to discuss and defend their design ideas. Thus these homeworks would become artifacts of their design process that would be included as part of their case study and overall semester project.

In order to provide students with more opportunities for collaborative interactions and learning through hands-on activities, our usability engineering course has also evolved from being a lecture-based course with many activities and a collaborative project to being a workshop. We have essentially ceased lecturing as of the 2006 version of the course; the students are involved in interactive activities in every class meeting. Our latest plan would orient this workshop more univocally on the semester project as an integrating workshop activity.

This instructional approach brings with it some inherent difficulties as evaluation of student learning is in many ways dependant on students' in-class discussions. In order for us to be able to evaluate how well students are grasping and applying core usability concepts we need to be able to peer into their thought processes. Rich, collaborative discussions are a useful way to accomplish this type of formative evaluation by drawing out students' ability levels while at the same time providing them with useful feedback. The problem we have encountered thus far is that students are not always comfortable with the risks involved with these sort of discussions: admitting they do not understand, applying concepts incorrectly, making themselves targets for criticism. The challenge for us will be to create a classroom environment where students feel it is okay if they do not have the "correct" answer as long as they can work with others to find it.

### B. *Plans for learning technology design*

Ten years after its initial development [Isenhour et al., 2001; Ganoe et al., 2003], BRIDGE is still a powerful collaborative environment. However, it is too complex for the usability engineering course application. It intimidates the students.

Beyond the specific requirements for distributed collaboration in the usability engineering course, we believe that Internet-based collaboration requires lighter-weight support that more easily integrates with various other Web-based services and information systems. Thus, we have begun to define and develop a new collaborative workspace for use in usability engineering course, and hopefully beyond it.

We think that a core functionality for such a system is the ability to collaboratively author documents. However, we do not think that it is necessary (or perhaps even desirable in some applications) to support character-by-character pushed synchronization, as we did in BRIDGE [Isenhour et al., 2001]. Thus, we are shifting toward asynchronous collaborative writing. Our approach is to allow collaborators to open and work on segments of a shared document. While a document segment is open, it is locked to other users. However, when saved, it becomes available to collaborators, who can open and continue to work on it. As we did in BRIDGE, we will support the coordination of such collaborative writing by maintaining an easily-accessible change history for document segments.

It is vital for students to master a range of collaborative skills in order to successfully accomplish complex collaborative projects. For this reason, we aim to scaffold important team processes such as planning, critical evaluation and reflection. We plan on accomplishing this through instructor modeling, hands on collaborative activities, reflective activities where students evaluate their team's processes to identify the team's collaborative weaknesses, and repeated practice in applying strategies to improve these weaknesses.

A system that supports education and learning should also be able to capture team process at a finer level in order to properly support the full range of collaborative competencies our students display. It needs to support planning, reflection, and evaluation of collaborative tasks and processes in order to allow instructors to help teams: diagnose problems, structure important collaborative processes, and support reflective activities centered on improving these processes. So in the collaborative workspace under design and implementation, besides collaborative writing service, we provide simple and flexible web services to support collaborative team-processes, such as meetings. Teams can create meeting agendas in a workspace and members can contribute by pooling their comments and thoughts as input; the output will be a meeting memo for their future reference. Teams can also create a shared to-do list by which they can track progress of their projects. For critical evaluation on their project design, students can create an idea pool to put their innovational ideas and do pros-and-cons analysis for each idea.

## V. CLOSING THOUGHTS

In this paper we discussed our experience with a particular assignment developed for our usability engineering course. We explained the main goals of the assignment and how it connected to the overall goals of the course. We also described the technological tools that could be used to support this activity. We then evaluated the activity after its implementation in order to identify ways that we could improve the activity and make it more useful for our students. Implementation and subsequent evaluation of the activity gave us better insights to our students and to the ways in which the activity was succeeding/failing to meet the original goals we set forth. This process of planning, implementing, evaluating, and refining our educational activity in many ways mirrors the iterative design process used in the development of technological systems. Indeed, it is only through the disciplined use of this design-feedback-redesign process that we can create useful and powerfully rich tools that can advance education and technology and the growing intersection of these two important disciplines.


## ACKNOWLEDGMENTS

Professor Mary Beth Rosson was co-investigator of the original UCS project, and led the original development of the project-based usability engineering course, the textbook for that course [Carroll & Rosson, 2002], and the Usability Case Study library. Professor Lu Xiao designed and developed the Collaborative Case Builder described in III-A. We thank our many usability engineering students at Virginia Tech and, at Penn State who have allowed us to study their learning activities during the past 14 years.


## REFERENCES


[1] Borge, M., & White, B. Y. (2009). Scaffolding collaborative processes with managerial roles. Paper presented at the American Educational Research Association, San Diego, CA.

[2] Carroll, J.M. 1990. The Nurnberg Funnel: Designing minimalist instruction for practical computer skill. Cambridge, MA: MIT Press.

[3] Carroll, J.M. (Ed.) 1995. Scenario-based design: Envisioning work and technology in system development. New York: John Wiley & Sons.

[4] Carroll, J.M. & Borge, M. 2007. Articulating case-base learning outcomes and assessment. International Journal of Teaching and Case Studies, 1(1/2), 33-49.

[5] J.M. Carroll, M. Borge, L. Xiao, and C.H. Ganoe, "Realistic learning activity is not enough", P. Diaz, Kinshuk, I. Aedo & E. Mora (Eds.), Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies: iCALT 2008 (Santander, Spain, July 1-5). Los Alamitos, CA: IEEE Computer Society, pp. 3-7, 2008.

[6] J.M. Carroll, and M.B. Rosson, "A Case Library for Teaching Usability Engineering: Design Rationale, Development, and Classroom Experience," ACM Journal of Educational Resources in Computing, 5(1), Article 3, pp. 1-22, 2005.

[7] Carroll, J.M. & Rosson, M.B. 2006. Cases as minimalist information. IEEE Transactions on Professional Communication, 49-4, 297-310.

[8] Ganoe, C.H., Somervell, J.P., Neale, D.C., Isenhour, P.L., Carroll, J.M., Rosson, M.B. & McCrickard, D.S. 2003. Classroom BRIDGE: using collaborative public and desktop timelines to support activity awareness. ACM UIST 2003: Conference On User Interface Software and Tools. New York: ACM, pages 21-30.

[9] Isenhour, P., Rosson, M.B. & Carroll, J.M. 2001. Supporting interactive collaboration on the Web with CORK. Interacting with Computers, 13, 655-676.

[10] Jiang, H., Ganoe, C. and Carroll, J.M. (Accepted) "Four requirements for Digital Case Study Libraries. " Education and Information Technologies.

[11] McMillan, D. (1996). Sense of community. Journal of Community Psychology, 24,315-325

[12] Microsoft, n.d., ASP Overview., retrieved: October 24, 2009 from http://msdn.microsoft.com/en-us/library/ms524929.aspx

[13] Rosson, M.B. & Carroll, J.M., 2002. Usability engineering: Scenario-Based Development of Human-Computer Interaction. San Francisco: Morgan-Kaufmann.

[14] Rosson, M.B., Carroll, J.M. & Rodi, C. 2004. Teaching Computer Scientists to Make Use. In I.F. Alexander & N. Maiden, (ed.), Putting Scenarios Into Practice: The State of the Art in Scenarios and Use Cases. John Wiley & Sons, pp. 445-463.

[15] Rosson, M.B., Carroll, J.M. & Rodi, C. 2004. Case Studies for Teaching Usability Engineering. Proceedings of 35th SIGCSE Technical Symposium on Computer Science Education (Norfolk, VA, March 3-7). New York: ACM Press, pp. 36-40.

[16] Rovai, A. P. (2002). Building a Sense of Community at a Distance. International Review of Research in Open and Distance Learning (On-Line), 3(1). Retrieved April 19, 2008 from http://www.irrodl.org/content/v3.1/rovai.html.

[17] Slavin, R.E. (1980). Cooperative learning. Review of Educational Research, 50, 2, 315-342.

[18] L. Xiao, J.M. Carroll, P. Clemson, and M.B. Rosson, "Support of Case-based Authentic Learning Activities: A Collaborative Case Commenting Tool and A Collaborative Case Builder," Proc. of HICSS 41: Hawaii International Conference on Systems Science, (Hilton Waikoloa Village, January 7-10, 2008). IEEE DL.

[19] Xiao, L., Clark, S., Rosson, M.B. & Carroll, J.M. 2008. Promoting reflective thinking in collaborative learning activities. In P. Diaz, Kinshuk, I. Aedo & E. Mora (Eds.), Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies: iCALT 2008 (Santander, Spain, July 1-5). Los Alamitos, CA: IEEE Computer Society, pp. 709-711.