Recognizing and Supporting Roles in CSCW

Mark Guzdial, Jochen Rick, and Bolot Kerimbaev
Graphics, Visualization & Usability Center
College of Computing
Georgia Institute of Technology
Atlanta, GA, 30332-0280
{guzdial, jochen.rick, bolot}@cc.gatech.edu

ABSTRACT

In this paper, we describe our experience with the long-term, widespread use of CoWeb, an asynchronous collaborative tool that is mostly used to complement existing face-to-face groups (such as classes). The CoWeb is an open-ended tool that does not enforce or explicitly support specific roles or usage, yet several well-defined uses and roles have emerged over time. In our design methodology, we recognize these roles and refine our collaboration environment to better support them.

Keywords
CoWeb, Wiki, CSCL, roles, website, long-term, design methodology

INTRODUCTION: DESIGN METHODOLOGY BASED ON ROLE IDENTIFICATION

In a traditional software methodology, tasks are assumed to be well structured, the domain well-understood, and thereby system requirements can be determined by a formal needs-assessment [12]. Creating a flexible collaboration tool that aims to be domain independent and offers the users opportunity to create their own structure does not fit well into this methodology.

Instead, we subscribe to an iterative design process. As a collaborative tool matures, new problems and phenomena appear. Initially, the main problem is getting the collaboration to happen. Next, the focus becomes supporting long-term and wide-spread use. Different roles emerge from this process. Finally, the developmental focus shifts to supporting these roles. This paper describes the roles we saw emerging in our work on CoWeb, and explains how we have refined CoWeb to better support these roles.

In January, 1998, we introduced the CoWeb (short for Collaborative Website), a new collaborative tool, to Georgia Institute of Technology (Georgia Tech). Introduction was made by simply mailing a few faculty who were interested in educational technology and offering them a URL where a sample CoWeb could be found. In that first academic term, almost 1000 students used the CoWeb across some half-dozen classes. Over the last two years, the growth of the CoWeb at Georgia Tech has been enormous. Over 120 CoWebs are now in use, with the majority of these being used to support course activities. We utilize ten servers to support this load. The uses and users of the CoWeb are diverse; CoWeb-using classes spread across the academic landscape of Georgia Tech, including Architecture, Chemical Engineering, Mathematics, English, Biology, and Computer Science [3, 5]. In these two years, we have had the opportunity to recognize and support a variety of roles associated with the CoWeb and its use.

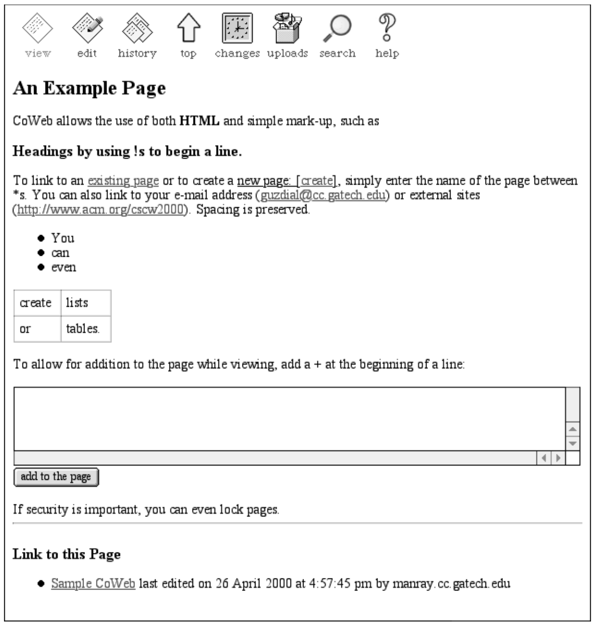
By roles, we mean specific concerns and activities associated with individuals who choose to use the CoWeb. Like Smith, Hixon, and Horan, we define a role as a human construct created and sustained by the interaction of minds [11], but our roles extend on and off-line and are not associated with capabilities within the shared space. Rather, like Stahl, we see learning as a rich social process, where it is the designers job to support and encourage these social processes [10]. Also, like Karsten and Jones, we find that the environment (CoWeb in our case; Lotus Notes in theirs) is used as a complement to face-to-face interactions and the overall interaction is changed by the affordances of the on-line environment [7]. So, the roles we see are products of the social process and the affordances of our environment. Thus, we can support specific roles without making them explicit in the environment.

Several of the roles we have identified are infrastructure roles, but unlike Star and Ruleder [12], our sense of roles are very explicit and are associated with individuals. We do not have a sense of individuals moving between roles as “levels,” but rather, that these roles are natural, stable (i.e., the role does not disappear after a period of time), and probably appear in most long-term CSCL (Computer Supported Collaborative Learning) and CSCW (Computer Supported Cooperative Work) applications. Our effort has been spent in recognizing, refining, and supporting the roles we have recognized.

THE COWEB AND ITS USES

The CoWeb is conceptually based on the WikiWikiWeb (or Wiki) by Ward Cunningham (http://c2.com/cgi-bin/wiki). The Wiki invites all users to edit any page within the website, and add new pages using only a regular web browser.

Figure 1: Viewing / Editing a CoWeb page in a Standard Web Browser



An Example Page

CoWeb allows the use of both **HTML** and simple mark-up, such as

Headings by using !s to begin a line.

To link to an existing page or to create a new page: [create], simply enter the name of the page between *s. You can also link to your e-mail address (guzdial@cc.gatech.edu) or external sites (http://www.acm.org/cscw2000). Spacing is preserved.

- You
- can
- even

create	lists
or	tables.

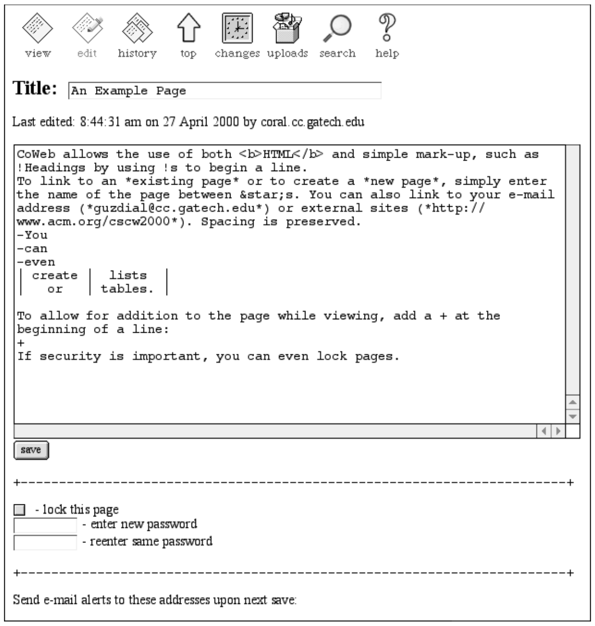
To allow for addition to the page while viewing, add a + at the beginning of a line:

add to the page

If security is important, you can even lock pages.

Link to this Page

- Sample CoWeb last edited on 26 April 2000 at 4:57:45 pm by manray.cc.gatech.edu



Title: An Example Page

Last edited: 8:44:31 am on 27 April 2000 by coral.cc.gatech.edu

```
CoWeb allows the use of both <b>HTML</b> and simple mark-up, such as
!Headings by using !s to begin a line.
To link to an *existing page* or to create a *new page*, simply enter
the name of the page between &star;s. You can also link to your e-mail
address (*guzdial@cc.gatech.edu*) or external sites (*http://
www.acm.org/cscw2000*). Spacing is preserved.
-You
-can
-even
| create | lists  |
|   or   | tables. |

To allow for addition to the page while viewing, add a + at the
beginning of a line:
+
If security is important, you can even lock pages.
```

save

+---+

☐ - lock this page
- enter new password
- reenter same password

+---+

Send e-mail alerts to these addresses upon next save:

The text is edited in an HTML text area without special applets or plug-ins. The Wiki is an unusual collaboration space in its total freedom, ease of access and use, and lack of structure. The Wiki is inherently democratic—every user has exactly the same capabilities as any other user. Where CoWebs strays from this principle is to assist when the natural social process has broken down. For instance, there is an administrator password that will allow the password holder to unlock pages; this is most often used when a novice user accidentally locks a page that is clearly communal property. For the most part, CoWeb allows for web collaboration without dealing with accounts and passwords. As one professor commented: "I just like the interaction that it enables. It's basically a whiteboard that everyone can write on. Protections are always kind of a pain." On top of being an interface problem, unequal capabilities influence which roles users fill rather than allowing them to develop through the social process; this is particularly important for roles that are ill-defined.

Like the Wiki, the CoWeb looks like a fairly traditional website, except that every page has a set of buttons at the top that allow the user to do various things such as edit the page, (un)lock the page, or view the history of the page over time. Links between pages are easily created by referencing pages within the same site by name (e.g., *Page Name*). If a page with the given name doesn't already exist, a *create* link shows up next to the name upon save; clicking on this creates the new page (See Figure 1).

The CoWeb differs from the Wiki in several important respects:

- The CoWeb is implemented in Squeak [6], a cross-platform and open-source freeware version of Smalltalk (Hence, the original name of the CoWeb, Swiki for "Squeak Wiki"). This has allowed us to use the CoWeb on virtually any server platform available, which has reduced one constraint on growth. The open-source freeware aspect of Squeak has encouraged community development and is somewhat responsible for the diversity of the roles we see.
- Where the Wiki provided a simple text notation but without HTML, the CoWeb supports both a simple text notation and HTML. The difference is due to our audience. In our academic community, most users know some HTML, and they want to use their prior knowledge.
- Unlike the more free-wheeling discussion-based context of Wiki, our use has been to encourage a specific context per CoWeb. Normally, each course has its own CoWeb. While CoWeb believes in equal power to all users, it is being used in contexts where there is someone "in charge" (the teacher or teaching assistants), whose words carry more weight than those of others, even if that person has no more additional capabilities than other users.
- CoWeb is easily adaptable. Many of our CoWebs contain extra features that were designed for those specific sites. For instance, two CoWebs allow users to easily share Matlab code. Other sites show "hot spots" at the top of every page.

Surprisingly, security has not been a problem in either the Wiki or the CoWeb, despite its open policy. Each has backup mechanisms in place that can be used for damage recovery, whether malicious or accidental, though little malicious damage occurs. We have even been struck by how well-behaved (e.g., fewer flames, more courteous language) students are on the CoWeb compared to the same students on a newsgroup. We hypothesize several possible reasons for this, though we have not studied this explicitly.

First, the public nature where anyone with a browser can visit may cause users to be more aware of their audience. Second, the persistent nature of the CoWeb (unlike a newsgroup, CoWeb pages are frequently revisited) may cause user to think of them as community property. Third, the web's growing status in our society may cause users to respect a web space more.

Originally, Mark Guzdial implemented CoWeb on top of Squeak's PWS (Pluggable Web Server). Jochen Rick redesigned CoWeb, which now works on top of the Comanche webserver by Bolot Kerimbaev. Comanche runs at 30% the speed of Apache for simple file serving. Like Artefact [2] and other tools to create dynamic web materials, PWS and Comanche support lightweight creation of artifacts. Served items can be created with embedded Squeak to ease the definition of dynamic materials. Unlike many other tools, however, Comanche lives inside of Squeak, so the creation of new kinds of served-artifacts is made easier through the Squeak IDE (Integrated Development Environment) and the many pre-defined multimedia objects available in Squeak. For example, a Comanche server module that would like to serve sounds or images need only return an instance of the Squeak classes SampledSound or Form. Conversion and appropriate MIME-typing is handled by Comanche and its support classes. This kind of automaticity eases the development of new kinds of dynamic web resources.

The uses of the CoWeb have varied dramatically across the different CoWebs. We identify four general categories of use:

- **Collaborative artifact creation**: In many classes, students using the CoWeb are collaborating to create some artifact. In some cases, the artifact being created is the website itself, with pages offering essays or particular information, such as discussions, that are created by students collaboratively. In other cases, the artifact is an analysis of a report or even a poem which is collaboratively marked-up and commented upon by the class. In still other cases, students are working on external artifacts (e.g., multimedia presentations) and use the CoWeb for coordination, planning, and distributing versions.
- **Review activities**: The CoWeb lends itself to sharing work and inviting comment. The reviewers may be internal (e.g., peer, students, teachers) or even external users. One of our most successful CoWebs was in Architecture where expert architects reviewed students' "pin-ups" on-line [14]. The very simple user interface of the CoWeb facilitated involvement by outside experts with little instruction and no specialized hardware or software. Instructors often use the CoWeb to comment on student's work and get students to comment on each other's work.
- **Case library creation**: CoWebs are frequently used as persistent spaces across offerings of a given course. As such, the contributions to the CoWeb of students in one offering of a course become information sources, and even advice, to later students. In some CoWebs, the creation of advice and information for future students becomes an explicit activity. Students post assignments, resources, and even "letters to future students" in the CoWeb to serve as cases for others.
- **Distributing information**: In our educational context, connecting the students with the instructor(s) and managing the class context are important activities. Because of the web environment, it is easy for users to link other web artifacts to the site for review or commentary. For some classes, connecting students to the professor has been critical to both the usefulness and the usage. In other classes, a shift of agency from teacher-student to student-student communication took place to the point that students agree that they learned as much from each other as from the teacher [4].

Figure 2 demonstrates these four uses of CoWeb with screenshots and descriptions of actual usage; the names have been changed (where possible) for anonymity.

Unlike tools such as EditThisPage.com or Microsoft FrontPage, the main purpose of the CoWeb is not to produce carefully crafted websites aimed at the casual web user. While the CoWeb can be used in this role, it is far more often used to involve visitors in the collaborative creation of the website. In our recognition of roles, we see author as being much more common than merely visitor, though we see both as important roles to support.

RECOGNIZING AND SUPPORTING ROLES

Over the course of the two years and over 15,000 CoWeb pages, we have recognized a variety of roles for our users, where the user may not always be someone directly using our software interface. These users may be students adding content, a teacher conducting an educational activity, or a systems administrator providing new pedagogical tools for the faculty she supports. As we have recognized roles, we have attempted to provide features and new tools to address the concerns and activities of these users. The below discussion describes the roles, in roughly the order in which we recognized them, and how we adapted our software to meet their needs.

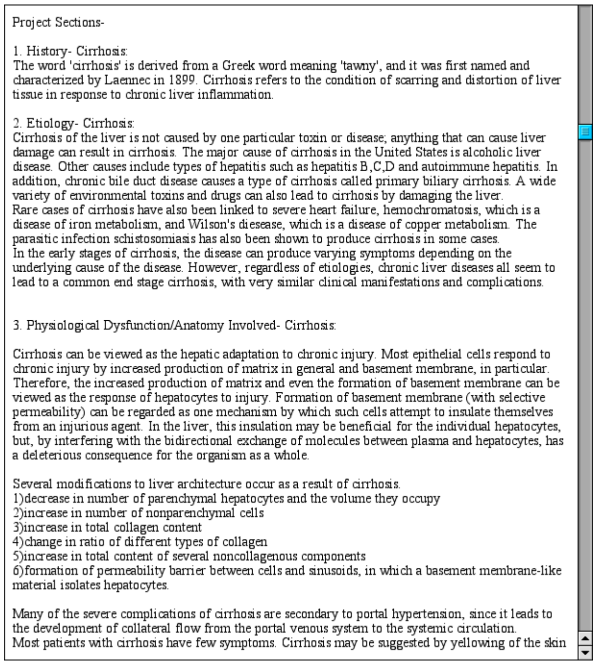
Authors

The most obvious role the CoWeb engenders (based on its foundation of the Wiki) is that of collaborative author. The main activity of a collaborative author is adding content (adding to existing pages or creating new pages). The author creates the collaborative artifacts, case libraries, and submits material for review; thus, he seeks to:

- find places to add content
- make sure that content is properly accessible
- find out where new content has been added
- create links between content (along the lines of Terveen and Hill's "emergent collaboration" [13])
- (as we discovered) recapture previously edited or modified content.

There were several features in the original Wiki that supported collaborative authors, and we merely copied them into the CoWeb. For example, a recent changes feature lists all pages in the site by the date on which they were last modified, most recent date on top; this makes it easy to find new or newly modified pages. Creating links by simply

Figure 2: Examples of CoWeb Usage (Screenshots and Explanations of Use)



Project Sections-

1. History- Cirrhosis:
The word 'cirrhosis' is derived from a Greek word meaning 'tawny', and it was first named and characterized by Laennec in 1899. Cirrhosis refers to the condition of scarring and distortion of liver tissue in response to chronic liver inflammation.

2. Etiology- Cirrhosis:
Cirrhosis of the liver is not caused by one particular toxin or disease; anything that can cause liver damage can result in cirrhosis. The major cause of cirrhosis in the United States is alcoholic liver disease. Other causes include types of hepatitis such as hepatitis B,C,D and autoimmune hepatitis. In addition, chronic bile duct disease causes a type of cirrhosis called primary biliary cirrhosis. A wide variety of environmental toxins and drugs can also lead to cirrhosis by damaging the liver.
Rare cases of cirrhosis have also been linked to severe heart failure, hemochromatosis, which is a disease of iron metabolism, and Wilson's diesease, which is a disease of copper metabolism. The parasitic infection schistosomiasis has also been shown to produce cirrhosis in some cases.
In the early stages of cirrhosis, the disease can produce varying symptoms depending on the underlying cause of the disease. However, regardless of etiologies, chronic liver diseases all seem to lead to a common end stage cirrhosis, with very similar clinical manifestations and complications.

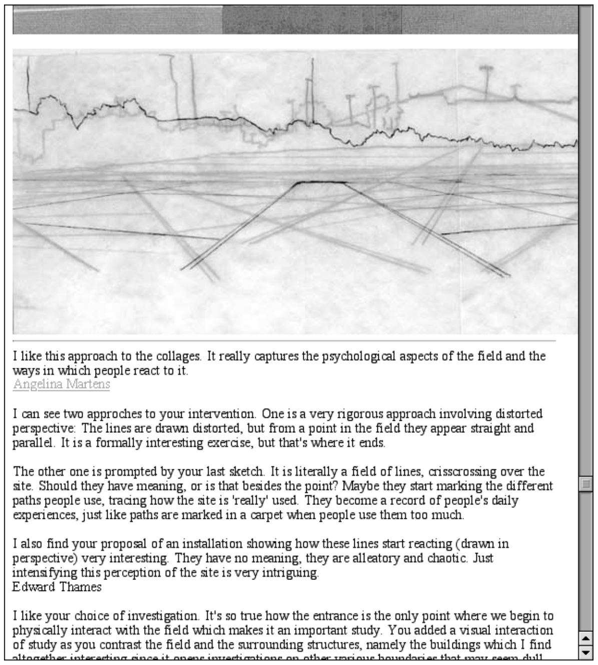
3. Physiological Dysfunction/Anatomy Involved- Cirrhosis:

Cirrhosis can be viewed as the hepatic adaptation to chronic injury. Most epithelial cells respond to chronic injury by increased production of matrix in general and basement membrane, in particular. Therefore, the increased production of matrix and even the formation of basement membrane can be viewed as the response of hepatocytes to injury. Formation of basement membrane (with selective permeability) can be regarded as one mechanism by which such cells attempt to insulate themselves from an injurious agent. In the liver, this insulation may be beneficial for the individual hepatocytes, but, by interfering with the bidirectional exchange of molecules between plasma and hepatocytes, has a deleterious consequence for the organism as a whole.

Several modifications to liver architecture occur as a result of cirrhosis.
1)decrease in number of parenchymal hepatocytes and the volume they occupy
2)increase in number of nonparenchymal cells
3)increase in total collagen content
4)change in ratio of different types of collagen
5)increase in total content of several noncollagenous components
6)formation of permeability barrier between cells and sinusoids, in which a basement membrane-like material isolates hepatocytes.

Many of the severe complications of cirrhosis are secondary to portal hypertension, since it leads to the development of collateral flow from the portal venous system to the systemic circulation.
Most patients with cirrhosis have few symptoms. Cirrhosis may be suggested by yellowing of the skin

This CoWeb page was used by students for **collaborative artifact creation**, where the artifact created was a summary overview of cirrhosis. The CoWeb allowed for the authors to easily access the information, be informed of changes, and add new content.



I like this approach to the collages. It really captures the psychological aspects of the field and the ways in which people react to it.
Angelina Martens

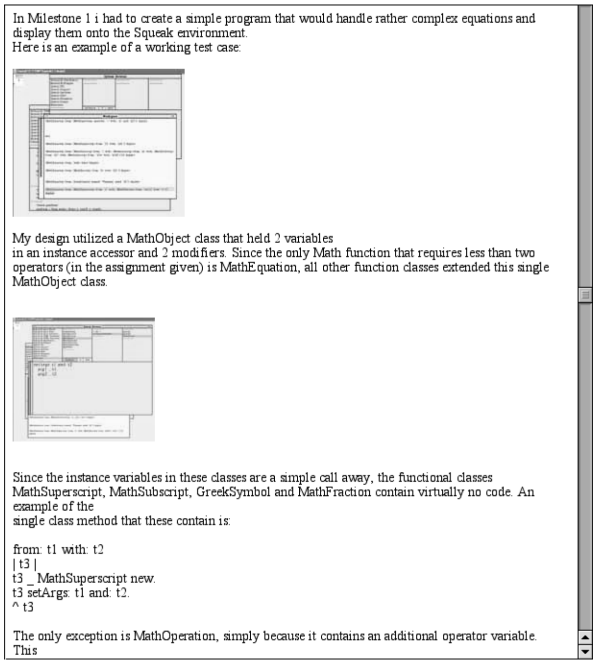
I can see two approches to your intervention. One is a very rigorous approach involving distorted perspective: The lines are drawn distorted, but from a point in the field they appear straight and parallel. It is a formally interesting exercise, but that's where it ends.

The other one is prompted by your last sketch. It is literally a field of lines, crisscrossing over the site. Should they have meaning, or is that besides the point? Maybe they start marking the different paths people use, tracing how the site is 'really' used. They become a record of people's daily experiences, just like paths are marked in a carpet when people use them too much.

I also find your proposal of an installation showing how these lines start reacting (drawn in perspective) very interesting. They have no meaning, they are alleatory and chaotic. Just intensifying this perception of the site is very intriguing.
Edward Thames

I like your choice of investigation. It's so true how the entrance is the only point where we begin to physically interact with the field which makes it an important study. You added a visual interaction of study as you contrast the field and the surrounding structures, namely the buildings which I find altogether interesting since it opens investigations on other various boundaries that may seem dull

In this CoWeb page, a student presented the work he had done for an architecture project by uploaded several pictures and sketches and explaining how these artifacts influenced his design. From there, class members (instructors, students, etc.) were able to **review** the work and give feedback.



In Milestone 1 i had to create a simple program that would handle rather complex equations and display them onto the Squeak environment.
Here is an example of a working test case:

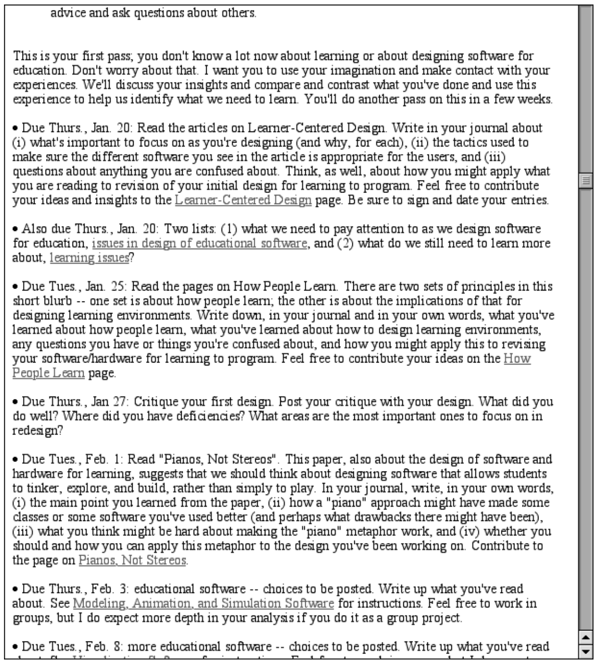
My design utilized a MathObject class that held 2 variables
in an instance accessor and 2 modifiers. Since the only Math function that requires less than two operators (in the assignment given) is MathEquation, all other function classes extended this single MathObject class.

Since the instance variables in these classes are a simple call away, the functional classes MathSuperscript, MathSubscript, GreekSymbol and MathFraction contain virtually no code. An example of the
single class method that these contain is:

```
from: t1 with: t2
| t3 |
t3 _ MathSuperscript new.
t3 setArgs: t1 and: t2.
^ t3
```

The only exception is MathOperation, simply because it contains an additional operator variable.
This

In this CoWeb page, a student in an object-oriented programming class uses the CoWeb for **case library creation**. For extra credit, she posts her project design and gives examples of working code. Other students will be able to use this case in future classes.



advice and ask questions about others.

This is your first pass; you don't know a lot now about learning or about designing software for education. Don't worry about that. I want you to use your imagination and make contact with your experiences. We'll discuss your insights and compare and contrast what you've done and use this experience to help us identify what we need to learn. You'll do another pass on this in a few weeks.

• Due Thurs., Jan. 20: Read the articles on Learner-Centered Design. Write in your journal about (i) what's important to focus on as you're designing (and why, for each), (ii) the tactics used to make sure the different software you see in the article is appropriate for the users, and (iii) questions about anything you are confused about. Think, as well, about how you might apply what you are reading to revision of your initial design for learning to program. Feel free to contribute your ideas and insights to the Learner-Centered Design page. Be sure to sign and date your entries.

• Also due Thurs., Jan. 20: Two lists: (1) what we need to pay attention to as we design software for education, issues in design of educational software, and (2) what do we still need to learn more about, learning issues?

• Due Tues., Jan. 25: Read the pages on How People Learn. There are two sets of principles in this short blurb -- one set is about how people learn; the other is about the implications of that for designing learning environments. Write down, in your journal and in your own words, what you've learned about how people learn, what you've learned about how to design learning environments, any questions you have or things you're confused about, and how you might apply this to revising your software/hardware for learning to program. Feel free to contribute your ideas on the How People Learn page.

• Due Thurs., Jan 27: Critique your first design. Post your critique with your design. What did you do well? Where did you have deficiencies? What areas are the most important ones to focus on in redesign?

• Due Tues., Feb. 1: Read "Pianos, Not Stereos". This paper, also about the design of software and hardware for learning, suggests that we should think about designing software that allows students to tinker, explore, and build, rather than simply to play. In your journal, write, in your own words, (i) the main point you learned from the paper, (ii) how a "piano" approach might have made some classes or some software you've used better (and perhaps what drawbacks there might have been), (iii) what you think might be hard about making the "piano" metaphor work, and (iv) whether you should and how you can apply this metaphor to the design you've been working on. Contribute to the page on Pianos, Not Stereos.

• Due Thurs., Feb. 3: educational software -- choices to be posted. Write up what you've read about. See Modeling, Animation, and Simulation Software for instructions. Feel free to work in groups, but I do expect more depth in your analysis if you do it as a group project.

• Due Tues., Feb. 8: more educational software -- choices to be posted. Write up what you've read

This CoWeb page was used by the instructor, of a class on educational technology, to **distribute information** (mainly class assignments). The CoWeb easily allowed her to post assignments with links to the relevant papers and to other parts of the CoWeb where students could share their work or begin related discussions.

typing a specially constructed phrase is a feature common to Wiki and CoWeb.

Though malicious damage was infrequent, accidental damage was common. A frequent cause was users overwriting one another's changes during heated contributions to a single page. From the very beginning, CoWeb stored all versions of a page. Initially, users sent e-mail messages to the administrator to get pages restored. Later versions of PWS CoWeb allowed users to access the last three versions of a page on-line, but even that proved insufficient. The latest version of CoWeb allows access to all previous versions of the page. In addition to keeping the complete history of a page, CoWeb also informs users if they are trying to overwrite a page with a version older than the saved version, asking the user to combine the two versions.

Since their work is openly accessible, some students fear that others might maliciously or accidentally wipe out their work. To ease these fears, we installed a locking mechanism that stops users without the password from editing the content (See Figure 1). If commentary on this work is still important, other users can be allowed to add to the page using the "Add to this Page" feature, but they cannot delete previous material.

Purpose Agents

For our use in education, another obvious role is the teacher. The concerns of a teacher are that students engage in useful learning activities with the CoWeb. Teachers give a purpose, or context, to the various activities that occur on the CoWeb. They may not be the most active users, but their actions greatly effect the use. Most activities that occur on the CoWeb are encouraged, or at least approved, by the teacher. To meet this end, teachers create a set of pages in a CoWeb to support certain activities: discussion pages, review-and-critique pages, and pages where students are expected to post work or personal pages. Usually, the teacher also uses the CoWeb to distribute class information to the students.

One of the concerns of teachers that we have learned to support is the creation of useful navigation. Students generally follow the structure of the CoWeb; however, a student-created case on the "Cases" page will probably not have a link from the case back to the "Cases" page, but teachers (and future students) would find such a link helpful. Thus, we added facilities to the CoWeb to automatically create back references: a list of links to the pages that contain references to the current page (See Figure 1). This feature automatically creates a useful set of navigation links which serve to make more useful the page structure created initially by the teacher.

Central Users

Because of the unstructured nature of the CoWeb and given that each user may have their own opinion of what that structure should be, CoWebs may become disorganized fairly quickly. To solve this problem, there is often a user who manages the meta concept of the structure. Their main interaction with the CoWeb is to guide the interaction of the authors to better structurally define the space. For instance, central users often integrate exceptional work into a case library, so that it is more prominently displayed. Although this role is sometimes filled by the teacher, it is more often done by others, such as teaching assistants or active students. In CoWebs where this role is not centralized to one person, the site often is hard to oversee, and redundant information flourishes.

To help the central users redefine the space, Comanche CoWeb allows users to change the title of any page without destroying the links to that page (See Figure 1). We have found that using the same CoWeb over several terms of a class is beneficial, as students are able to use the previous class's work as cases by which to model their own work. Being able to change the page titles allows the central user for these sites to more easily merge the "new term" context with the old space; for instance, the old "Homework Set #5" page can be changed to "Fall 1999: Homework Set #5."

Since the central user has not been predetermined, they are often self selected. These users sense the necessity of the role and have enough social power and technological familiarity to fill the position. The emergence of a central user is similar to Bonnie Nardi's emergence of local developers in programming communities [8]. They appear because there is a social need for their existence. Nardi noted that the function of the informal local developer can also be replaced by a formal permanent position [8]. In several of what we consider successful CoWebs, there are formal central users (such as teaching/research assistants).

Peripheral Users

Two roles that emerged on the CoWeb over time were those of viewers (i.e., visitors who were just viewing material but not contributing) and reviewers, who were often external to the University and were offering advice to students. Whether these roles emerged depended strongly on the specific context of that site. In particular, Architecture classes consider it valuable for students to present their work to external critics for review [14]. Often, students invited their friends and family to look at their personal page(s). The peripheral users were invited by a member of the core (central users, purpose agents, and authors) users; central users and purpose agents invited reviewers, while authors invited viewers. In contrast to the other roles, the peripheral users were supported mainly through the actions of the core users instead of the developers. As developers, our focus becomes creating features that better enable the core users to support the peripheral users.

For visitors, we, as developers, added nothing explicitly to support them; supporting navigation for authors also supported viewing concerns and activities. Supporting reviewers proved a bit more difficult; reviewers are infrequent (peripheral) visitors who do not want to learn much to contribute. To ease the learning curve and support easier collaboration, we added three features.

- Reviewers and authors could annotate a page with their e-mail address to be informed whenever the page changed. Reviewers could use this feature to receive notice when students (or other reviewers) responded to their comments.

- The central users with the help of the administrator could set up some templates that changed the open-ended text area to a prompted multi-field submission. Thus, users could be scaffolded along the process of reviewing and creating documents for review.
- Authors could add special notations that created small text areas and "Add to page" buttons that enabled additions to the page without fully editing the page (See Figure 1). This has proven to be useful in enabling viewers and reviewers to contribute without facing a whole page of source.

Site Designers

As the diversity of uses of the CoWeb grew, a standard look-and-feel was found insufficient by some of the users. For example, in one class studying a short story that was later made into a movie, the teacher wanted to incorporate elements of the movie into the interface on every page. Site designers want to tweak the "look and feel" of their site, without modifying all other CoWebs on the same server, and with minimal exposure to underlying issues of HTTP and Squeak code. However, manipulating HTML was acceptable and even expected by these designer.

From the earliest form of the CoWeb, the "look and feel" of a CoWeb was defined in terms of template files that were mixed HTML and dynamic tags. These template are evaluated at run-time to translate them into the HTML that was actually served to the client. Asking site designers to edit the template files was acceptable to them. They were able to easily understand what was happening, and were able to modify the HTML around the dynamic tags. But, as the CoWeb has gained features, the number of templates increased and modifying templates became more tedious. Current CoWebs offer three facilities to ease this burden:

- CoWebs have an inheritance hierarchy associated with them, so that the "look and feel" can be defined as a composition of features from existing CoWebs. This allows for a certain customization without dealing with HTML or Squeak.
- Simple modifications to the looks can be made over a web interface. For instance, the image directory, which is responsible for the icons at the top of the page, can be pointed to any web directory. Thus, the site designer could work on these images in their web space without having to have access to the hosting machine.
- We are creating utilities that will ease the editing and development of new CoWeb interfaces.

Developers

The CoWeb has proven itself useful at Georgia Tech as a general mechanism for collaboration, besides its role in CSCL. As such, it has been used as collaboration support for other projects. For example, CoWeb has been integrated with the Classroom 2000 project for capture-and-retrieval of lectures in order to create a collaboration and discussion space off Classroom 2000 [1, 9]. In another project, the CoWeb was used to create a "collaborative radio;" users upload sound files and "play lists" which were used to define broadcasts on a small FM transmitter. In both of these cases, the flexibility of and access to the underlying structure was critical. The CoWeb merely provided a collaborative framework; the contents of those documents or the indices to them were manipulated by software external to the CoWeb.

Because of the open-source freeware aspect of Squeak, other people outside of our group have been able to develop applications using the CoWeb infrastructure. To satisfy these developers, an open framework was necessary that could easily allow for modification. This framework has proven effective. For instance, one external developer was able to translate the CoWeb from English to German by modifying the configuration files and without modifying the core software.

Administrators

As the CoWeb grew in popularity across campus, interest in setting up new servers grew. In some cases, departments wanted to set up their own servers for their faculty. The more common case was a faculty member wanting to use his desktop computer to serve CoWebs.

The cross-platform nature of Squeak made it possible to offer CoWeb serving on virtually any platform on the faculty member's desk, but the fairly arcane knowledge of Smalltalk required to set up the server initially was a limiting factor. The administrator does not want to be an expert on the underlying technology; he does not want to know about HTTP, special directories, or other issues of concern to someone setting up a traditional webserver, like Apache. The administrator just wants to know how to start/stop the server, create new CoWebs, and find out their URLs.

Over the last two years, we have created a succession of documentation and of administration utilities to meet the concerns and activities of this role. The important goal has been to make Squeak virtually invisible to the administrator. Setting up and configuring the CoWeb is done entirely through the web browser, which is what the administrator really cares about; he simply starts the server executable and goes to the admin(istrator) CoWeb to manage his site.

Support Staff

Far before the tenth webserver went into use, we realized the complexity of our own role in supporting multiple CoWeb servers, each serving multiple CoWebs. Our concern was for the stability, maintainability, and robustness of the servers, and our ability to respond to problems. Our concerns were, in some sense, completely separate from issues of the CoWeb itself, yet our role was to be expected when a technology like the CoWeb becomes as pervasive as it has at Georgia Tech.

There are various technologies that are useful for these purposes. For some of our concerns, we used existing technologies; for others, we developed our own. We created a "Swiki Spy" to regularly monitor the servers and report any problems. CoWebs can inherit features and properties from other CoWebs; this aided not just the site designers but also the support staff, as it allowed us to modify just a few CoWebs during maintenance and upgrade, allowing the improvements to trickle down to all the other CoWebs on that site.

Table 1: Roles, their Concerns, and Features to Address those Concerns

Role	Central Concerns and Activities	Supporting Features or Tools
Authors	Add material, find new material, connect related material, manage the accessibility of that material	Monitoring features, support for composition and reclamation
Purpose Agents	Encourage proper usage and specify the usage context	Navigational support
Central Users	Structure the space	Re-classifying material
Peripheral Users	Contribute easily	Notification and annotation support
Site Designers	Change the end-user interface's "look and feel"	Templates and customization support
Developers	Apply the technology to new applications and easily add new features	Powerful language, easy to use API (Application Programmer Interface)
Administrators	Administer the site without knowing the technical details	On-line administration utilities
Support Staff	Support the maintenance and robustness of several servers	Monitoring and upgrade supports

CONCLUSIONS: SUPPORTING DIFFERENT ROLES IN AN OPEN ENVIRONMENT

As our roles are largely social, we expect that they are common in the growth of collaborative technologies, and our strategies for meeting the needs are transferable to other settings (see Table 1). We are encouraged in this position by two findings. First, Smith, Hixon, and Horan identify similar roles in Kansas (a synchronous small-group collaboration tool) as we identify here in CoWeb (an asynchronous large-group collaboration tool) [11]. Second, Nardi notes similar definition of roles in end-user programming of spreadsheets (a very different kind of technology) [8]. As collaboration is becoming more prevalent in both education and the workplace, new tools will be necessary to enable this interaction. CoWeb is such a tool, albeit asynchronous and only supporting on-line interaction. We predict that many of our roles will transfer to other collaborative scenarios (non-educational and non-academic contexts, different synchronicity, and other environments).

Although some of the roles that we found were obvious to us going in (authors, support staff), others (central users, peripheral users) did not appear until later in the software development and usage. Other findings, such as the importance of a central user and the divergence of purpose agent and central user surprised us. Using the CoWeb in different domains, creating new applications with it, and evolving the software in an open-source community has given us an insight into what the needs of a wide variety of users are. Adding new features to meet some of their needs has allowed even more users to become involved. Most features we created ended up serving more than one need across several roles and even generated new usage. That usage in turn created the opportunity for other users to emerge. By constantly having user feedback, we were able to develop the software to fit the needs of the actual users, some of whom we would not have envisioned without this development cycle. This iterative design methodology allowed us to evolve a flexible open-ended collaboration tool that is quite popular and works in a variety of contexts.

ACKNOWLEDGMENTS

This research is supported by a grant from the National Science Foundation, REC-9814770, an AEG grant from Sun Microsystems, a grant from the Mellon Foundation, and seed funding from Siemens and the GVU Center at Georgia Tech.

The authors would like to thank Joshua Gargus, Colleen Kehoe, and other reviewers for their commentary and insight.

REFERENCES

1. Abowd, G., Pimentel, M. d. G., Kerimbaev, B., Ishiguro, Y., & Guzdial, M. (1999). Anchoring discussions in lecture: An approach to collaboratively extending classroom digital media, *Proceedings of CSCL'99* (pp. 11-19). Palo Alto, CA.

2. Brandenburg, J., B. Byerly, T. Dobridge, J. Lin, D. Rajan and T. Roscoe (1998). Artefact: A framework for low-overhead web-based collaborative systems. *Proceedings CSCW98* S. Poltrock and J. Grudin. Seattle, Washington, USA, ACM: 189-196.

3. Craig, D., ul-Haq, S., Khan, S., Zimring, C., Kehoe, C., Rick, J., & Guzdial, M. (2000). Using an unstructured collaboration tool to support peer interaction in large college classes. Paper presented at the *International Conference of the Learning Sciences 2000*, Ann Arbor, MI.

4. Guzdial, M., Realff, M., Ludovice, P., Morley, T., Kerce, C., Lyons, E., & Sukel, K. (1999). Using a CSCL-driven shift in agency to undertake educational reform, *Proceedings of CSCL'99* (pp. 211-217). Palo Alto, CA.

5. Guzdial, M. (1999). Supporting Learners as Users. *The Journal of Computer Documentation*, 23(2), 3-13.

6. Ingalls, D., T. Kaehler, J. Maloney, S. Wallace and A. Kay (1997). Back to the Future: The Story of Squeak, A Practical Smalltalk Written in Itself. *OOPSLA'97 Conference Proceedings*. Atlanta, GA, ACM: 318-326.

7. Karsten, H. and M. Jones (1998). The Long and Winding Road: Collaborative IT and organisational change. *Proceedings CSCW98*. S. Poltrock and J. Grudin. Seattle, Washington, USA, ACM: 29-38.

8. Nardi, B. (1993). *A Small Matter of Programming: Perspectives on End User Programming*. Cambridge, MA, MIT Press.

9. Pimentel, M. d. G., Y. Ishiguro, B. Kerimbaev, G. D. Abowd and M. Guzdial (2000). "Supporting Long-Term Educational Activities through Dynamic Web Interfaces." *Interacting with Computers* (Accepted, To Appear).

10. Stahl, G. (2000). A Model of Collaborative Knowledge-Building, Proceedings of ICLS2000 (pp. 70-77). Mahwah, NJ.

11. Smith, R. B., R. Hixon and B. Horan (1998). Supporting flexible roles in a shared space. *Proceedings CSCW98*. S. Poltrock and J. Grudin. Seattle, Washington, USA, ACM: 197-206.

12. Star, S. L. and K. Ruhleder (1994). Steps toward an ecology of infrastructure: Complex problems in design and access for large-scale collaborative systems. *Proceedings CSCW94*. R. Furuta and C. Neuwirth. Chapel Hill, NC, USA, ACM: 253-264.

13. Terveen, L. and W. Hill (1998). Evaluating emergent collaboration on the Web. *Proceedings CSCW98*. S. Poltrock and J. Grudin. Seattle, Washington, US, ACM: 355-362.

14. Zimring, C., S. Khan, D. Craig, S.-u. -. Haq and M. Guzdial (1999). CoOL Studio: Using simple tools to expand the discursive space of the design studio. Design Thinking Research Symposium, MIT, Cambridge, MA.