Designing Online Environments to Facilitate Classroom Management and Student Collaborative Work

Approved by

Dissertation Committee:

[redacted]

John Slatin

[redacted]

Margaret A. Syverson

[redacted]

John Ruszkiewicz

[redacted]

Allucquére Rosanne Stone

[redacted]

Cynthia Selfe

Designing Online Environments to Facilitate Classroom Management and Student Collaborative Work

by

Michael Murray Chorost, B.A., M.A.

Dissertation

Presented to the Faculty of the Graduate School of

the University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

December 2000

Learning is but an adjunct to ourself
And where we are our learning likewise is.

William Shakespeare, *Love's Labour's Lost*

Acknowledgements

Grateful thanks go to my dissertation advisor, John Slatin, who provided superb direction and feedback while I plowed through tens of thousands of words of misguided drafts and dead ends. This is a far better dissertation than it would have been without his thoughtful, thorough, and informed guidance.

Special thanks go to Peg Syverson for her course on pedagogy in electronic environments, and for excellent advice at many points during the designing of the Collaboration Center and the writing of this dissertation. Many thanks, also, to the rest of my committee, John Ruszkiewicz, Sandy Stone, and Cindy Selfe, for helping me through this rite of passage.

In many ways, this dissertation builds on the pioneering work done by my predecessors and colleagues in the CWRL. In his history of computer-facilitated instruction at the University of Texas at Austin, Fred Kemp (1996) notes that he and his peers had to wire their computers together by hand and write their own low-level networking software in DOS. By contrast, my work on the Collaboration Center was able to take advantage of the servers, systems administrators, and high-speed Internet connections that they carefully built up over the years. Equally importantly, I was able to build on the institutional structures and theoretical vocabulary they helped to build. The CWRL can now support large instructional computing projects as a main research focus, instead of as heroic skunk-works projects conducted under the academy's puzzled gaze. And in terms of theory, I inherited a vocabulary for talking about the mutual synergy of teaching and computing, which they pioneered in building in the late 1980s and 1990s. I am

deeply grateful for the years of work they put in to create a robust foundation for their successors.

Thanks to Brett Elliott, ex-sysadmin extraordinaire of the CWRL, who supported the installation of ColdFusion and helped me with numerous technical issues over four semesters.

Thanks to my colleagues in the CWRL, especially Joanna Wolfe, Bob Hornback, Lynn Rudloff, Wade Rowland, Susan Warshauer, Bret Benjamin, and Darren Cambridge, from all of whom I learned much in conversation and observation.

Thanks to the teachers who used the Collaboration Center, including Lisa Hernandez, Paige Schilt, Daniella Mallinick, David Barndollar, Brian Bremen, Phillip Barrish, and most especially, Jan Tucker.

Thanks to my colleagues at Scient in San Francisco, Angela Arnold, Richard Charette, and Rico deLara, and to Gary Bridges and Joseph Jones at SRI International, for continued moral support.

Thanks to two my two terrific cats, Elvis and Friday, who always welcomed me home with love and demands for dinner.

A grateful tip of the coffee cup goes to Kenny, Daniel, and other employees at the Jim's Restaurant on Route 183 in Austin, where much of the early work on this dissertation was done during long nights working at a table at the back of the restaurant. They fed me when I was hungry, talked to me when I needed company,

and let me work undisturbed during stretches which often lasted from 8 p.m. to 3 a.m. Jim's Restaurant is quite possibly the best place in the world for graduate students to get work done.

Thanks to my friends David Chioni Moore, Jeff Weill, Julia Shuford, and Alane Fitzgerald for longtime advice, friendship, and support during the long, arduous years of graduate school. And thanks, also, to each of these people plus Amanda Swinghamer and my father, Sherwood Chorost, for last-minute emergency help in reformatting the bibliography from MLA to APA format. Thanks, also, to Meg Worley of Stanford University, who capably proofread an earlier draft of this dissertation.

Thanks are due to Compaq for their donation of a 400-MHz Pentium II server with 256 megabytes of RAM to the CWRL in 1998. The Collaboration Center lived on this server for two semesters, with much greater speed than it had while running on my student account on an overburdened UT server

I want to acknowledge the support of Gallaudet University, which awarded me a graduate student fellowship in 1992 and 1993.

Thanks of course go to Mom and Dad, whose labor pains started with my birth and have never, I think, really stopped. They supported me unfailingly through my long pilgrim's progress through graduate school. I was also helped in many ways by my grandparents, Sam and Edythe Leibowitz. Sadly, my grandfather died in July 2000, just short of seeing this dissertation completed.

And of course, I offer many and deep thanks to my students, who were happy to collaborate in a long and productive experimentation with new modes of teaching, learning, and communicating.

Designing Online Environments to Facilitate Classroom Management and Student Collaborative Work

Publication No. _____

Michael Murray Chorost, Ph.D.
The University of Texas at Austin, 2000

Supervisor: John Slatin

This dissertation is about the design and use of online tools to facilitate classroom management and student collaborative work. It draws from management theory, complexity theory, and constructivist pedagogy to propose that classrooms are information ecologies where communications and resources flow in many directions. Given the complexity of such environments, online tools can play a valuable role in facilitating communication, coordination, and teamwork. The dissertation discusses a test bed software project, the Collaboration Center, which was developed over four semesters to experiment with ways of using online tools to manage classrooms and support student teams. It draws from this experience to propose design principles which can aid in the creation of a next generation of classroom management and team collaboration software.

Table of Contents

Introduction 1

1. The Scene of Learning 5
   The Course 5
   The Students 7
   The Classroom 9
   The Collaboration Center 11
2. What This Dissertation Is Not About 14
3. What This Dissertation Is About 16

Chapter 1: Managing the Constructivist Classroom 17

1. The Ecology of Learning 17
2. Management Theory: Complexity in Leadership and Learning 24
3. The Properties of Complex Systems: Locality, Topology, Ecology 29
   Locality 29
   Topology 37
   Ecology 38
4. The Limitations of the Ecological Metaphor 47
5. Family Resemblances: Complexity Theory, Constructivism, Network Pedagogy 49
6. Network Pedagogy in the Technological Classroom 51
7. Managing Complex Learning Environments with Software 57

Chapter 2: The Collaboration Center 63

1. Introduction 63
2. Relational Databases and Classroom Management 65
3. Integrating Groupware With Pedagogy 71
4. The Evolution and Content of English 309M 73
5. A Functional Description of the Collaboration Center 78
6. Using the Collaboration Center to Support Learning in Complex Environments 83
   Locality 85
   Network Topology 96
   Maintaining an Ecology of Knowledge 121

Chapter 3: Design Principles for a Next Generation of Collaborative Learning Environments 138

1. The Agenda of Software Design 138
2. The Meaning of "Design" and "Use" in a Social Context 141
   Design and Metadesign 141
   Social Orientation as a Success Factor in Software Implementation 144
3. The Limitations of Software 146
   Practical Limitations 147
   Theoretical Limitations 148
4. Designing to Afford Community Systems Design 149
   Examples of Design 151
   NegotiationLens 153
   ActionTechnologies' ActionWorks Metro 153
5. Examples of Affording Community Systems Design 158
   SenseMaker 158
   ActiveInk 160
   How Do Current Web-Based Programs Measure Up? 162
6. Four Design Principles for a New Generation of Groupware 166
   Simplicity/Flexibility 167
   Spatiality 172
   Modularity 178
   Extensibility 181
7. Conclusion 189

Appendix 1: Answers to Surveys on Activity Tracking 191

Appendix 2: The Syllabus of English 309, Fall 1999 199

Bibliography 205

Vita 234

Introduction

This dissertation is about how I designed, created, and used online tools to manage constructivist, team-oriented learning in the undergraduate composition classes I taught between Fall 1997 and Spring 1999. Since constructivist pedagogy emphasizes open-ended discussion and teamwork, it stands to benefit from tools which make such activities easier and richer. I discuss how I designed and used the Collaboration Center, an integrated suite of 28 Web-based tools, to enrich student, teacher, and team communication during and outside of class sessions.

The discussion is situated within a constructivist perspective on learning and teaching, which is elaborated in Chapter 1. Chapter 1 draws from three literatures – management theory, chaos theory, and constructivist pedagogy – to make the case that learner-centered, constructivist classrooms can be regarded as information ecologies, where knowledge is collectively built through complex, multilateral communications among teachers and students. Such learning environments are hard to manage by their nature, since constructivist pedagogy decentralizes the classroom, taking the locus of control away from the teacher. I argue that to manage constructivist classes effectively, teachers would benefit by taking two interrelated courses of action: first, by adapting techniques which professional managers have developed to handle decentralized environments, and second, by learning to use automated, online tools to manage the complexity of the interactions that go on in teams. The first course of action is discussed within Chapter 1; the second course of action is discussed in Chapter 2.

In Chapter 1, I suggest that since management theory has long dealt with challenges such as setting goals, organizing teams, setting up communications flows, and resolving conflicts, it has much to offer pedagogy. In the constructivist

shift to team-oriented and project-based work, both teachers and students increasingly take on the roles and responsibilities characteristic of knowledge workers in professional environments. For example:

- Students and professionals are both set tasks which they must achieve collaboratively.
- Students face analogous challenges of scheduling, negotiating, decisionmaking, and workflow, albeit on a smaller scale.
- Students' need for distance-collaboration technologies is becoming more like that of professionals, since campuses are larger, and more students work (reducing their time on campus).
- Because of the Internet, the two groups have a more similar technological base than ever before, e.g., email clients and Web browsers.
- Teachers and supervisors confront the increasingly similar problems of how to design collaborative environments, both in material and online spaces.

Teachers, then, are likely to become increasingly like managers, a shift which in places I acknowledge by using the tag "teacher/manager" instead of the word "teacher" alone – this denotes a teacher who views teaching as inseparable from management. I am not, however, suggesting that teachers aim to turn their classrooms into imitations of corporate life. Scholastic values differ substantially from professional ones. Universities regard education as a public good with deferred, long-term benefits, whereas professionals usually require it to have some concrete and quantifiable economic return (Allen, 1999). What I am suggesting,

rather, is that teachers can critically examine how organizations have approached the problems of collaborative work and selectively adapt promising solutions.

Chapter 2 is about how the Collaboration Center was designed and used to manage the complexity of multilateral communication and teamwork. While obviously not all communication and management tools need to be online ones, I focus on online solutions because they take advantage of a powerful communications technology – computer networks. Networked computers are excellent recordkeepers and distributors of information. When computing is used as a communications medium among students and student teams, the accumulated information about their interactions can provide a window onto their activity and production. Software can provide a supporting infrastructure of multilateral communication, information storage, and event coordination.

The Collaboration Center's tools served a variety of functions. Some of the tools, such as the announcement spaces and the Knowledge Base, enabled students and teachers to provide information to each other. Other tools, such as the Forums and the team rooms, supported team activity by allowing students to post, discuss, and coordinate work. Still other tools, such as the attendance log, the syllabus, and the "latest events" report, supported the administrative needs of the course by making it easier to manage collective activity. Chapter 2 discusses innovative uses and worthwhile discoveries made with each of these kinds of tools.

A principal difference between the Collaboration Center and earlier software experiments lies in its scope. In the 1980s and early 1990s, researchers explored the possibilities of programs which enabled students to converse online, particularly forums, MUDs, and real-time chat. These programs, powerful as they

were for their time, were nonetheless mainly restricted to facilitating synchronous textual communication and offered little scope for managing the course as a whole. They could offer users only the most minimal information about student activity: logins and transcripts of chat sessions, if that. The Collaboration Center, on the other hand, offers communication tools *and* tools for overseeing and administrating classroom and team activity, in one integrated package. It offers, among other things, tools for surveys, appointment scheduling, file storage, attendance recording, syllabus maintenance, and team formation. It can display detailed records of user activity to both students and teachers, giving a vivid sense of social life in an online space. It is a *place* where students and teachers come and go, doing various things of their own accord at all hours of the night and day. That enables this dissertation to talk about using software to manage class activity as a whole, taking in its scope activity in the physical world as well as online. That is why this dissertation aims to enlarge the focus of the conversation from *enabling discussion* to *managing classroom activity*. Working life in the next century will increasingly be intertwined with online spaces, taking advantage of their ability to represent activity and facilitate it. Chapter 2 is aimed at helping the educational infrastructure evolve similarly.

Chapter 3 is a speculative and prescriptive chapter. It builds on the lessons learned from the Collaboration Center, and from my observations of other software packages, to derive four principles to guide the construction of a next generation of collaborative learning software. These are, in brief, simplicity/flexibility, spatiality, modularity, and extensibility. I argue that a next generation of collaborative software needs to offer tools which are simple and flexible enough to permit ready improvisation of classroom social processes, and provide participants with a rich visceral sense of the presence of the other participants. Teachers should

be able to assemble a selected variety of tools within a single interface, and programmers should be able to build new tools relatively easily. Software designed from this perspective will offer teachers a rich set of resources which they can bring to bear in designing constructivist learning environments.

1. The Scene of Learning

The scene of learning in English 309, the course I taught over four semesters, was both material, that is, a four-walled physical classroom, and virtual, that is, the online space created by the Collaboration Center. Here I will briefly describe the course, the students, the classroom, and the Collaboration Center; each of these will be discussed in more detail in the dissertation.

The Course

English 309 is an elective undergraduate writing course with approximately 25 students of all majors and years. Each section is unique, in that it has its own topic and its own syllabus. For example, there have been English 309s on romance writing, travel writing, and autobiography. I took a different approach in my section of English 309; thematically, it was about the process of collaboration itself. The required readings were about team writing, group dynamics, and communications techniques. (The syllabus for the Spring 1999 class is given in Appendix 2.) The rest of the students' reading was self-chosen and project-related. Since student teams had considerable latitude in choosing the topics of their projects, their readings were on a wide range of topics, from genetic engineering to the *Titanic* sinking to the Microsoft trial.

As befits a constructivist writing course, students usually worked in teams, with rotation into new teams for each project. Most of the classroom time was spent working on team projects, presenting Knowledge Workshops (see below), discussing projects and readings as a class, and observing teacher technology demonstrations. Relatively little time was spent using the Collaboration Center during class itself, except peripherally by students to check items such as the syllabus, the class announcement, and their team's project room as class was starting. Virtually all student use of the Collaboration Center was outside of class.

The key assignments of the course were as follows:

Initial team projects. The semester started with a series of three brief (2-3 day) team assignments. Students were cycled into a new team each round, enabling them to get to know roughly half their classmates before launching into the major projects.

Research papers. These were exercises in historiography and collaborative writing in paper documents. Students were asked to choose a topic, a particular magazine, and a three-year time period fifty or more years ago, and write a paper detailing how the periodical dealt with that topic during that period.

Knowledge Workshops. Students were asked to poll their peers (using the Collaboration Center) to determine a topic of interest to a significant minority of the class. Working in teams of two or three, students designed a complete lesson and taught it to the class.

Web Debates. The Web Debate consisted of a collaboratively created Web site, emphasizing multiplicity and difference instead of consensus in writing. Teams of students were asked to research different positions on a topic of their own choosing and build "branches" of a Web site to put them in juxtaposition with each other.

Web Final Project. Students were required to create their own web site as a final project, on a topic of their choice. The assigned size was between 20 and 30 nodes, which required careful organization and good interface design.

Final portfolios and self-evaluations. At the end of the course, students submitted portfolios along with a two- to three-page self-evaluation. The object was to encourage reflection upon the student's learning trajectory and consolidate the experience of the course.

The Students

As usually happens in elective courses, each of my four English 309 classes was different from the last in its composition of genders and majors. In the first two semesters, Fall 1997 and Spring 1998, the course was divided roughly 50-50 between men and women, with a wide range of majors. In the second two semesters the gender balance shifted in favor of men, with three women out of 21 students in Fall 1998 and five out of 25 in Spring 1999.[1] In the Spring 1999 course, fourteen of the 25 students were computer science majors. In that semester, as in all of them, I took especial care to reassure the non-CS students that their Web

[1] The course rosters may easily be viewed by guests. Readers of this dissertation may log into the Collaboration Center by going to http://corax.cwrl.utexas.edu/cc and logging in with a last name of "Dissertation" and the password "phd". This grants access roughly equivalent to a student's.

work would be judged on organization, content, and team participation, not on technical facility.

In fact, I saw no correlation between academic majors and the quality of students' Web project organization and content. Once students learned the basics of using Netscape Composer to build Web pages and WS_FTP upload them to server accounts, they were at no disadvantage in terms of creating well-organized HTML navigation schemes and writing good online content. For their final project, they were also required to include at least one image which they had scanned and edited by themselves using Photoshop, and this skill enabled them to produce sites with original pictures and artwork. I repeatedly told the non-CS majors, "This course is about working with people to create well-organized, well-written Web content, not about technology. Once you master the basics of Web site authoring, you have as good a shot as anyone at getting an A." Creating simple Web sites is not difficult, as the history of the Web amply demonstrates, and all of my students were able to learn the basics regardless of their major or technical background. Many of them expressed pleasure, surprise, and relief after going through the course's (optional) workshop, which led them, step by step, through building and uploading a simple Web site: "That's all there is to it?"

The majority of my students were of typical undergraduate age and were Texan by birth (according to the "Hometown" entries in the Roster). The majority were Caucasian, with a significant minority (9 out of 25 in the last semester) of Asian descent and a smaller minority of Hispanic and African-American students.

In the summer of 1999, I interviewed ten of my former students to ask them about their memories of the course and the Collaboration Center. These interview

comments are shown in indented Times Roman font and identified as interview comments in the text. Written student comments (e.g. in online surveys conducted in the Collaboration Center and in end-of-semester surveys) are shown in Courier font. These are exact transcriptions of what they wrote.

The Classroom

My section of English 309 met twice weekly, Tuesday-Thursday, in a classroom in the basement of the university's undergraduate library, the Flawn Academic Center. This room contained 25 computers. These were set against the walls, arranged so that the monitors faced the room's center (see Figure 1). A collection of six rectangular tables occupied the center, arranged to form a hollow square. With this geometry, students could face outward to work at the computers, or face inward for class discussion. For small-group work, I had the students split up the rectangular tables and distribute them around the room, each group to its own table. The room was thus highly configurable for class needs.

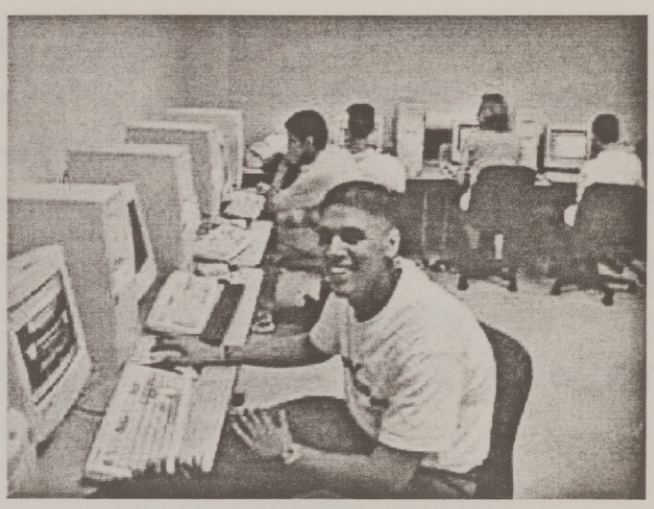


Figure 1: FAC (Flawn Academic Center) 9, the classroom where my courses were taught. All four walls had computers arranged along them in this fashion. (Photo courtesy of the Computer Writing and Research Lab).

The computers were Gateway 266-MHz Pentium IIs with 64 megabytes of RAM and high-bandwidth connections to the Internet. Each machine had Microsoft Office and a full complement of Internet software, including Netscape and Eudora (for email). Additional equipment included a scanner connected to a machine with a copy of Photoshop, and an inFocus projector, which I used frequently to display Web sites or application software for demos and discussions. The classroom (along with four others) was supported by a full-time systems administrator, an administrative specialist, numerous lab staffers, and several high-capacity servers. The Collaboration Center itself ran on a Compaq Pentium II-400 server with 256 megabytes of RAM and a six-gigabyte hard disk, donated by Compaq in an educational grant program.

The out-of-class computing environment, that is, the computers owned personally by the students, was arguably even more important than the in-class one,

since most of the usage of the Collaboration Center occurred outside of class meeting times, as was intended. Virtually all of my students owned Pentium-class or better computers with 56k or better connections to UT's Internet service provider. Of the 100 or so students who went through the course over four semesters, only a handful – perhaps five or six – did not own their own computers. Even these students, however, had UT computers readily available to them. Thus the course was taught in the context of a technologically saturated computing environment with generally computer-literate students.

The Collaboration Center

The Collaboration Center grew over four semesters from a single module (the Roster) to a collection of 28 integrated ones. The following screenshot shows a "typical" view of it, from a student's viewpoint. The student sees 18 functions in a toolbar at the top (teachers and administrators see a larger subset). This particular page is the page shown after the student logs in, welcoming him or her to the site and showing some personal information plus the class announcement.

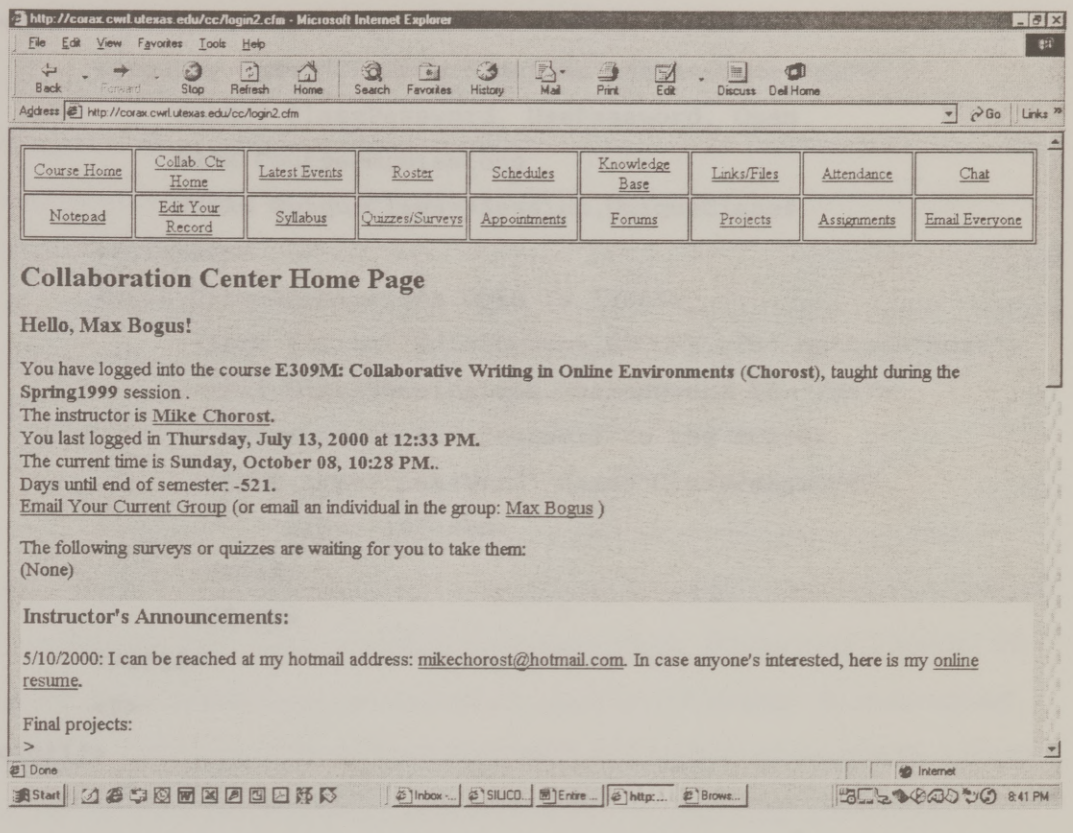


Figure 2: The entry page of the Collaboration Center. It welcomes the user with information about the course, and the instructor's announcement. This particular screenshot was taken long after the course had ended; this is why the page reports that there are "minus 521" days until the end of the semester.

Every page the user sees in the Collaboration Center is dynamically generated from a database stored on the server, with code written in ColdFusion, a Web development language. In all, there are roughly 19,000 lines of code, written over four semesters of development. When I say that I "developed" the Collaboration Center, I mean that I did all of the design, programming, and database work that brought it into being. I spent many, many hours creating and debugging code that looked like this:

```
<cfif Session.AccessLevel IS Application.Guest>
      <cfquery name="ClassPermissions" datasource="cc">
            SELECT ClassPassword, UsePassword, Open
            FROM CoursePermissions
            WHERE UniqueClass=#Session.UniqueClass#
      </cfquery>
      <cfif ClassPermissions.Open IS "Yes">
             <form action="EditStudent_SECURE.cfm" method="post">
             <cfif ClassPermissions.UsePassword IS "Yes">
               Password to Add Yourself to the Roster
             <input type="password" name="ClassPassword"
                   size="10"><br>
             </cfif>
             </form>
      </cfif>
      <P>
</cfif>
```

This work was a far cry from the normal classroom routine of a writing teacher. It was even further from the high-level theoretical concepts of constructivist pedagogy and management theory which I was aiming, at the same time, to understand and apply. But the code slowly accumulated into software which enabled my students and I to experiment with new and innovative forms of communication and work. I added new modules in direct response to classroom needs, and my students and I evolved new uses for existing modules. In addition, the course itself changed, often substantially, to take advantage of new capabilities in the software. Thus there are few, if any, simple causal relationships to describe. The software did not develop in direct linear relationship to the course, nor did the course activities and goals spring linearly from the software's capabilities.

For this reasons, this is a very different dissertation than would have been written had I used a commercial package such as Lotus Notes, the Daedalus Integrated Writing Environment, or CommonSpace. That dissertation would have been about *using* software. This dissertation, on the other hand, is about being both a creator of and a user of that same software, which is a richer, more recursive, and more revealing process. It was an extremely labor-intensive way to teach a course, but it afforded a richer interplay between software design and pedagogical goals than would have been possible with commercial software.

Why did I not use a commercial product? For one thing, none were available in 1997. Lotus Notes, the only plausible option at the time, was difficult to install and required an expensive proprietary client. For another, I wanted the power of being able to design tools to my exact specifications. As it turned out, this has enabled me to write about the interplay of design and usage with a degree of richness and insight that I could not have acquired by using commercial software.

2. What This Dissertation Is Not About

It is important to establish what this dissertation is *not* about. It is not about distance learning, although my experiments and findings may have relevance to educators interested in designing distance learning programs. The Collaboration Center was intended to enrich rather than replace the face-to-face classroom experience. Even though the majority of accesses to it were outside class meetings, these did not make my course into a distance learning experience, any more than assigning homework would have done. Just as my students did homework on their own time, they also accessed the Collaboration Center on their own time.

It is also not about collaboratively composing texts inside a shared interface. My students composed with word processors and Web authoring programs, and shared their work by annotating and merging files. In this respect, the Collaboration Center differs from programs such as CommonSpace, which enables participants to asynchronously post and comment on texts, and NetMeeting and WebEx, which enable real-time, mutually visible annotation and manipulation of texts within a Web browser.

Finally, it is not about delivering instructional materials online. The Collaboration Center offered only rudimentary functions for delivering instructional materials, limited to the simple transmission of text files online. In this respect it differs from the products of vendors such as DigitalThink, which deliver prepackaged and relatively interactive subject-matter courses on well-defined topics such as programming languages and operating systems. For this reason, this dissertation does not engage critics of technology like David Noble (1988), who fear that computing stands to disempower academics by taking control of knowledge out of their hands:

> Most important, once the faculty converts its courses to courseware, their services are in the long run no longer required. They become redundant, and when they leave, their work remains behind…Today faculty are falling for the same tired line, that their brilliance will be broadcast online to millions. Perhaps, but without their further participation (Online).

While I share Noble's concern about preserving the rights of faculty to their intellectual property, these particular issues were not at stake in my creation and use of the Collaboration Center. It did not deliver a set body of knowledge. To put it another way, a visitor to the Collaboration Center would be able to acquire very

little factual knowledge about the subject being taught, apart from downloading whatever text files had been posted to the class's Knowledge Base. On the other hand, she would be able to learn a great deal about the class itself: its students, its discussions, its teams, and those teams' ongoing projects.

3. What This Dissertation Is *About*

What this dissertation *is* about, on the other hand, is extending a maturing paradigm of constructivist teaching and learning in technological environments. It aims to bridge constructivist pedagogy and management theory; it proposes expanding computer-facilitated pedagogy's scope to managing classroom activity as a whole; and it bases its principles of software design on an understanding of collective activity as an ecological phenomenon, calling for environmental design, facilitation, and improvisation rather than control.

Constructivist teaching is not an easy thing to do, since it rejects the Taylorite efficiencies which make lecture-oriented teaching so apparently simple to administer and evaluate. Teaching in technological environments is also challenging, because it requires hard-won familiarity with software and hardware, and a finely tuned sense of how to integrate them seamlessly into the life of the class. Doing them *together* squares the difficulty. This dissertation aims to help the profession succeed with both of these simultaneous challenges, by offering theoretical foundations and concrete lessons learned from my four semesters of teaching using the Collaboration Center.

Chapter 1: Managing the Constructivist Classroom

1. The Ecology of Learning

A computer-facilitated, learner-centered classroom is, in some ways, like a living organism. It has a life of its own. Students meet in small groups, talking among themselves, occasionally rushing off to the nearest computer to find some resource or to start composing. Projects grow and take shape. The instructor, a source of perturbation, influences but can hardly control the activity and output of each group. The outside world, swirling about them, has its own perturbing effects: other courses, roommates, the party last Thursday, boyfriends and girlfriends, distant families. In this environment, the instructor is more like a gardener than an architect, perturbing and channeling the restless energy in the room rather than controlling it. As Margaret Syverson has shown, collaboratively written student papers emerge from an ecology of many interacting influences: the assignment, the instructor's feedback, the physical setting of composition, the peculiarities of computer hardware and software, and the physical characteristics of the writers' bodies (Syverson, 1999). This dissertation is about managing learning in such environments, with the aid of computer networks to facilitate communication and collaboration.

The terminology of complex systems is increasingly being used to describe social organizations. In their book *The Complexity Advantage: How the Science of Complexity Can Help Your Business Achieve Peak Performance* (1999), Suzanne Kelly and Mary Ann Allison suggest that businesses can be thought of as complex systems. They go so far as to compare them to living organisms by suggesting that they are *autopoietic*, in the sense that "an autopoietic system is a self-organizing system that creates its own boundaries and preserves and renews itself over time"

(p. 28). They support this claim by pointing out that teams and departments form in response to environmental needs and change over time as circumstances perturb them.

The claim that organizations can be described in the same terms used for biologically living organisms needs to be treated with caution. The term "autopoiesis" was originally invented by Maturana and Varela (1980) to distinguish between the kind of complexity that characterizes biological organisms and that which characterizes social collectives, economies, and languages. According to Terry Winograd and Fernando Flores (1987), biologically living systems have a special *kind* of complexity, characterized by an ongoing re-creation of internal morphology and function:

> An autopoietic system holds constant its organization and defines its boundaries through the continuous production of its components. If the autopoiesis is interrupted, the system's organization – its identity as a particular kind of unity – is lost, and the system disintegrates (dies) (pp. 44-5).

Social systems meet part of this definition but not all of it. The behavior of biological organisms arises without centralized, rationalized, intentional control. Biological organisms exhibit *emergent* behavior patterns; that is, the behaviors emerge out of a decentralized interaction among their components. The behaviors are a property of the system as a whole. According to Margaret Syverson (1999),

> Emergent properties have been proposed as as the counter-effect to entropy: they are tendencies toward the self-organization, order, and structure that emerge from simple components that might be expected to exhibit either random, chaotic behavior, or stable, predictable behavior – but which in fact, do neither. Yet this order is not created or determined by a single central master "executive" or "brain" (p. 25).

Social systems are strongly conditioned by intentional and rational control, but they are also complex systems which exhibit some degree of emergent behavior. They are complex in a richer sense than a designed artifact such as an engine or a computer chip. A computer chip is merely *complicated*, but a biological organism is *complex*. A system that is complex exists "at the edge of chaos," where, according to Waldrop, "the components of the system never quite lock into place, and yet never dissolve into turbulence, either" (Waldrop, 1992, p. 12). This kind of complexity permits stability and change to coexist together, giving an organism that protean adaptability which characterizes living systems.

If one drew a continuum between the kind of complexity that a computer chip has and the kind of complexity that a biological organism has, social organizations would lie somewhere in between. Like biological organisms, they have a loosely coupled and (to some extent) decentralized structure; but like designed artifacts, they also have a rationalized structure and organization. Their properties are both designed and emergent. They are both complicated and complex.

In this dissertation, my interest is in the complexity that arises in a classroom run along constructivist principles. While such a classroom is surely not as complex as a living organism, it is not merely complicated, either. As the opening paragraph of this chapter suggested, students interact in surprising and unpredictable ways, and the particular projects that are chosen and the topics that arise cannot be predicted from the class's initial conditions (e.g. the syllabus, the students' majors, and the teacher's intentions.) The rational intent of the actors

(that is, the teacher and the students) has only partial control over the course's outcome, for three reasons.

- Human organizations are composed of many and not always agreeing human actors.
- Complex systems are influenced by artifacts and external conditions beyond intentional control.
- Humans have limited ability to comprehend complex systems.

This is why I will draw on some insights from complexity theory to describe the interactions in a constructivist classroom and to propose computer-based management tools for managing them effectively.

Complexity theory was originally proposed by physicists, mathematicians, and economists to model loosely coupled systems with large numbers of actors following well-defined rules. They define it in mathematical and computational terms, and those definitions are used to model entities such as weather patterns, population shifts, and stock markets (Gleick, 1993; Kauffman, 1995; Waldrop, 1992). As the diversity of those entities shows, complexity theory is interdisciplinary; for example, it has influenced cognitive science's models of the mind (Hofstader, 1979; Winograd & Flores, 1987), biology's definition of life (Maturana & Varela, 1980), and even cosmology's account of the origin of the universe (Smolin, 1997).

Complexity theory has become heuristically valuable in socially-oriented disciplines such as sociology, management theory, and pedagogy. Studies in these fields have focused on producing a nuanced description of how various entities

(people, artifacts, intentions, and circumstances) have interacted to produce an observed outcome, such as a ship's navigation into port (Hutchins, 1995), the production of student papers and literary texts (Syverson, 1999), a corporate librarian's research deliverables and the failure of a hospital monitoring system (Nardi & O'Day, 1999), student participation in a classroom newsgroup (Guzdial, 1997), and a business' competitive advantage (Allison & Kelly, 1999).

As these examples suggest, a complexity-theory perspective shifts the focus of the analysis from individual actors to the system. Management theorists Allison and Kelly (1999) regard this displacement as axiomatic: "And, contrary to what we may have thought in the past, the billions of interactions in the business enterprise cannot be consciously constructed and controlled by leaders, entrepreneurs, and managers" (p. 25). More succinctly, Edwin Hutchins (1995) writes, "The larger system has cognitive properties very different from those of any individual" (p. 226). Thus despite the difference between social-scientific projects and those of physicists and mathematicians, they have at least one crucial postulate in common: that system behavior is not describable solely in terms of organizational intentionality. This postulate focuses attention on the cognitive properties of the system as well as on the individual actors in it. For pedagogy, the goal becomes to arrive at deeper understanding of the dynamics of information ecologies, in order to help teachers manage them in the classroom and to prepare students for the complex environments which lie ahead of them.

Constructivist pedagogy has focused on the "larger system" by drawing on social psychology (Vygotsky, 1978) and studying classroom conversational dynamics, both face-to-face and online (Forman, 1994; Kiesler, Siegel & McGuire, 1988; Slatin, 1992; Warschauer, 1998; Wolfe, 1999, 2000). Recent theories of

classroom management have characterized the teacher as a manager, environment designer, and facilitator (Bennett, 1999; Bruffee, 1993; DeLong, 1997; Elbow, 1973; Hillocks, 1995; Johnson, Johnson, Holubec, & Roy 1984; Lemke, 1996; Varn, 1999). In this way constructivist pedagogy has moved closer to management theory in becoming intensively concerned with building environments to support teamwork and collaboration.

Network technology accelerates this trend. It has often been proposed that computer networks can reform education by helping classrooms become sites of collaborative knowledge creation, not just knowledge transmission (Cox, 1998; Harasim, 1996; Koschmann, 1996; Scardamalia & Bereiter, 1996). The emerging pedagogical paradigm is that classrooms, in shifting to project-based and computer-facilitated collaborative work, will become more like workplaces, and that teachers will consequently become more like managers.

The similarities are further enhanced by the fact that classrooms and professional workplaces now have access to many of the same kinds of information technologies. In computer-facilitated classrooms, campus computer labs, and on their own computers, students encounter essentially the same infrastructure they will see in professional environments: email clients, browsers, and office suites. John Slatin (1992) has suggested that the far-reaching changes Shoshana Zuboff documented in industry will now, in their turn, be seen in the classroom, brought about by the textualization of oral and physical modes of of communication (p. 46). Among the changes that Zuboff (1988) perceived and foresaw was a greater fluidity and complexity of relationships between actors:

> The relationships to be managed are both more dynamic and more intricate than earlier patterns. The shape and quality of relationships will vary in

> relation to what people know, what they feel, and what the task at hand requires. Relationships will need to be fashioned and refashioned as part of the dynamism of the social processes, like inquiry and dialogue, that mediate learning (p. 400).

Slatin's analysis of his classes' chat sessions bears out Zuboff's findings. They reveal "a shift away from the standard pattern of initiation-response-evaluation that dominates traditional classroom talk," with student-to-student and student-to-class messages accounting for the majority of conversational traffic (p. 42). Slatin's role as teacher was not to direct the class to a particular conclusion but rather to "provide information, synthesize student comments, or refocus the discussion" (p. 42). This is a facilitative role in a dynamically complex environment of multilateral interactions. Kevin Kelly (1995) suggests that the appropriate term for this kind of management is "co-control" (p. 331). The teacher co-controls, but does not control, the classroom environment.

Other teachers have also documented this shift. Mark Warschauer (1998) has shown how incorporating e-mail into a writing course had the effect of decentralizing classroom discourse and giving students greater autonomy. The teacher had initially insisted on treating the email messages as if they were essays to be read and graded before sending, but the the students resisted, feeling that the exercise put the emphasis on form rather than communication. Halfway through the semester, the teacher relented and allowed them to correspond freely with their partners, a change much welcomed by the students (p. 30). It is particularly interesting to note that this happened at a small fundamentalist college in Hawaii with a scant tradition of learner-directed education. This is an example of how Internet technology acts as a Trojan horse, introducing new values into environments which might have resisted them otherwise (see also Guzdial et al., 1999).

Given these reconfigurations in the nature student work, the dynamics of classroom discourse, and the increasingly managerial role of teachers, it makes sense to ask what management theory can contribute to pedagogy.

2. Management Theory: Complexity in Leadership and Learning

In order to bridge the fields of management theory and pedagogy, I argue that complexity theory offers the clearest descriptive theory of how modern organizations work, and the most effective prescriptive theory of how teaching can prepare students to function effectively within them. It enables me to ground my teaching on a model of decentralized, team-oriented interaction, where teacher-led activity is deemphasized and students are required to define their own goals and methods to a considerable extent. I envision my role as that of environment builder and standards-setter, not as direct controller of activity. As part of my role as environment builder, I created software to help manage the complex and multilateral communications that go on in a distributed organization. The bulk of my dissertation is devoted to describing how I implemented this pedagogy on a day-to-day basis, and how my software supported it.

Complexity theory's influence is evident in the work of many management theorists (Davenport, 1997; Drucker, 1998; Fisher & Fisher, 1997; Kelly and Allison, 1999; Nonaka, 1998; Senge, 1990). It grounds their prescriptive theories of how leadership works and how organizations should be structured. All of them posit that the small team is the basic unit of the organization, and they prescribe "environmental" strategies for helping teams work more productively. For

example, Thomas Davenport (1997) favors "information federalism" as a rubric for organizational governance:

> Federalism involves representative democracy, a weak central government, and a high level of local autonomy. In information terms, this model emphasizes that only a few information elements need to be defined and managed centrally, while the rest can be left up to the local units (pp. 68-69).

Management theory emphasizes the local focus of the team and the situatedness of knowledge in a particular context and community. James Bair's definition (1997) of knowledge management (KM) focuses on "leveraging" the individual's powers of action:

> KM is an emerging set of processes, organizational structures, applications, and technologies that aim to leverage the study of the capable, responsible, autonomous individual to act quickly and effectively. KM achieves this end by providing this capable, responsible, autonomous individual with ready access to the enterprise's entire store of knowledge, including much of what is known but not documented (p. 4).

Bair's definition focuses on process, individual autonomy and responsibility, comprehensive access to relatively heterogeneous information, and the situatedness of information (e.g. "tacit" knowledge). He notes that KM is likely to fail in environments where autonomy and access are formally or informally limited:

> Enterprises with cultures that systematically limit or inhibit capability, autonomy, and responsibility, as well as those in which sharing of knowledge is actively discouraged either by official or unofficial policy, will find that investment in KM technology provides (relatively) minor operational efficiencies at best (p. 5).

Many organizations have emphasized the word "learning" over "teaching" as a way of encouraging this situatedness and self-directedness. According to Bijan Masumian (personal communication, March 1997), a manager at AMD, the word "training" misrepresents their educational department's mission on two counts. First, it implies the transmission of rote skills, which belie the high-level knowledge that employees needed to acquire in order to manage and improve chipmaking technology. Second, it implies a managerially driven process, in which management tells employees what and when to learn. On the contrary, employees frequently define their own needs and seek the department's help in meeting them. Thus "learning" better represents their work, because it puts the focus on the learner and the work situation rather than on the teacher and classroom.

Complexity theory also grounds descriptive theories of organizational workings. While she does not use the language of complexity theory, Rebecca Grinter's meta-analysis of CSCW (Computer-Supported Collaborative Work) research reveals a focus on the dynamism and nonlinearity of workplace environments (1997). She notes that the following observations appear repeatedly in the literature:

- People make assessments about data based, in part, on the status of the provider.
- Individuals make some of their work visible to others, and also monitor each other.
- People's perception of technology affects the ways that they use it.
- Work is dynamic and involves many channels of communication.
- Spatial information provided by the arrangement of papers and personnel lets others know the current activities of the entire group.

- People construct and share interpretations of the work in progress.
- Work often deviates from the planned activity in order to accommodate situated action.
- Maintaining context supports long-term collaboration (p. 232).

Corporate ethnographers such as those working in CSCW seek to understand the flow of information and the network of entities and relationships through which work gets done. They often pay relatively little attention to intentionalist structures such as organizational charts and verbal accounts – indeed, they often assume that they are irrelevant if not actively misleading (Fellman, 1999; Suchman, 1987; Underhill, 1999). A related field, activity theory, seeks to describe the outcomes of the interactions of many different actors (Engeström & Escalante, 1996; Fuhrer, 1996; Lave, 1996; Lynch, 1995).

Thus academic and industry theorists often propose theories and solutions which, structurally speaking, have much in common. The techniques discussed by academic theorists such as Timothy Koschmann, John Slatin, Mark Warschauer, Donald Morrison and Bruce Goldberg, and Christine Neuwirth and Patricia Wojahn closely resemble those proposed by management theorists such as Wanda Orlikowski, Peter Drucker, Peter Senge, and Kimball and Mareen Fisher. Both industry and academic solutions focus on ill-defined domains; both privilege distributed communication over vertical hierarchy; both dissipate concentrations in space and time; both expand the public sphere at the expense of the private; both privilege *habitus* over rules (Bourdieu, 1977, p. 78); and both look to technology as a fertile ground of solutions.

The similarities between constructivist pedagogy and management theory do not, however, mean that educational and management theorists are necessarily thinking of the same things when they think about learning. Schools and organizations have different goals and responsibilities, and different criteria for measuring success. One key difference between school and organizations is that in the latter, education is often not perceived as a separate category at all. Theorists of industry learning are often little focused on classroom training. Rather, they are focused on what Peter Senge (1990) calls "the learning organization," that is, the organization whose continued existence is directly dependent on continuous learning, dialogue, and adaptation. In rapidly changing environments, with mutable product lines and diverse client needs, organizations which do not build learning into their fundamental structure become uncompetitive and risk irrelevance. They are likely to characterize educational issues in managerial terms rather than in pedagogical terms (e.g., "How do we manage knowledge?" and "How do we bring new teams up to speed quickly?") They are also likely to measure success in more quantitative terms than academic theorists, e.g., the speed with which products are brought to market, or the number of clients successfully served in a unit time. Thus for industry theorists of learning, education is so embedded in daily practice that it may not appear as a category at all.

Academic educators, on the other hand, are more focused on learning as a final goal. But even there, educators are increasingly seeking to simulate the environments that characterize learning organizations, bringing themselves in closer alignment with industry modes of education. The classroom is no longer simply a site of transmission; it is a site of activity and engagement with authentic challenges. It is becoming less important to master a fixed body of information than it is to become able to adopt its practitioners' ways of speaking about it and

extending it. For the purposes of this dissertation, the similarities are more important than the differences.

3. The Properties of Complex Systems: Locality, Topology, Ecology

In this dissertation I focus on three key properties of complex systems: locality, topology, and ecology. They are not the only properties of complex systems, of course, but they are the most useful ones for describing how I designed the Collaboration Center and analyzed its usage.

- *Locality* refers to the primacy of local functional units – in this case, teams and individuals – as the principal site of work.
- *Topology* refers to the fact that the elements of complex systems typically relate to each other through networks rather than hierarchies.
- *Ecology* refers to the fact that ecological and organic metaphors work better as descriptors of complex systems than mechanical metaphors.

In the rest of this section, I'll discuss each of these in turn, relating them briefly to how I attempted to implement them in a classroom setting. More detailed discussions follow later in this chapter and in Chapter 2.

Locality

Complexity theory privileges local functional units as the basis of analysis and development, instead of master plans, hierarchies, or totalizing narratives. By contrast, the "contemporary language of work" emphasizes master plans and totalizing narratives:

> We remain, in the final years of the twentieth century, prisoners of the vocabulary in which managers require employees; superiors have subordinates; jobs are defined to be specific, detailed, narrow, and task-related; and organizations have levels that in turn make possible chains of command and spans of control (Zuboff, 1988, p. 394).

Complexity theory offers a very different account of the relationships between actors in a system. Douglas Hofstader's account of cognition (1979) is a well-known example. He offers a theory of mind as a decentralized, self-organizing collection of complexly interacting "symbols." Hofstader does not commit himself to any particular neurological account of what symbols are, but he suggests that they are patterns of electrical activity moving through the brain, acquiring and losing strength as they interact with other symbols. Their local interactions are the components of higher-level interactions, which are themselves still local in their interactions. One can move up the hierarchy of functional descriptions, describing the brain in increasingly less physical and more symbolic terms, until one arrives at the highest level of functional description: the intentional, self-aware, feeling, thinking mind.

Since each level has nothing *other* than its constituents with which to interact, the elements in them are are both constitutive of and constituted by their environment. This makes the model deeply self-reflexive, since a symbol's activity changes the very environment with which it interacts; it becomes caught up in what Hofstader calls a "Strange Loop."

This self-reflexivity renders the search for a center or origin in vain. There are no "extra-smart" neurons, nor ethereal soul for that matter, envisioning and purposively designing the outcome. Each individual functional unit of the brain, be

it physical like a neuron or virtual like a signal, acts without purposive awareness. This makes it possible to describe the workings of the mind in functional rather than teleological terms. It is only at the top level that teleology becomes important, where one has to speak of the mind in terms of intentionality. Below the top level, the mind can be described in terms of its functional units, on any of many different levels.

Hofstader's model of mind can be viewed as emblematic of models of other complex systems, including living organisms, human organizations, ecologies, and economies. The mapping is not precise, but the basics of the model can be carried over to each: the lack of a central, controlling, intentional entity; the mutually constitutive interactions of many parts; the ability to describe the whole on different functional levels; the option of describing the system without relying on intentionality; and the assumption of interaction with an environment suffused with information and resources. All of these rest on the idea of locality.

Models of complex systems iterate simple rules upon a large number of locally-acting entities. Each entity acts on its own, with no conception of the larger whole, yet global consequences and patterns emerge. Stuart Kauffman (1996) suggests that locality may be a key factor in the success of large organizations:

> The results hint at something deep and simple about why flatter, decentralized organizations may function well: contrary to intuition, breaking an organization into "patches" where each patch attempts to optimize for its own selfish benefit, even if that is harmful to the whole, can lead, as if by an invisible hand, to the welfare of the whole organization (p. 247).

How can individual selfishness – that is, locality – be reconciled with organizational success? According to Kaufmann, local units *coevolve* through their mutual interaction. The prosperity of local areas changes the environment of the areas around them, and can increase the propensity of those areas to succeed as well. If the founding circumstances are right, successful solutions are propagated through the whole:

> We are about to see that if the entire conflict-laden task is broken into the properly chosen patches, the coevolving system lies at a phase transition between order and chaos and rapidly finds very good solutions (pp. 252-3).

The problem, of course, is to know what those founding circumstances are, and to explain how they arise without intentional design. Kaufmann acknowledges that this question has not yet been fully answered, but he believes that the properties of matter and energy are such that they will often settle spontaneously into coevolving, order-generating systems – "order for free," as he calls it. He suggests that a mature theory will aid in setting up organizations whose local units are likely to coevolve toward effective solutions (p. 266).

The work of Kaufmann and the Santa Fe group of theorists has created the beginnings of a theoretical grounding for locality within a larger theory of complexity. However, the concept of locality is no longer a remote theoretical concept; it actively informs practical planning and analysis. A 1999 *New York Times* article notes how organizational life has been increasingly influenced by a Hayekian model of information exchange, which Hayek developed in contrast to socialist models of centralized power. According to Hayek (1945), centralized organizations are unable to process the enormous amount of information emerging out of a technological economy, and thus cannot respond fast enough to its

changing circumstances to manage it effectively. Hayek's solution is to devolve power to smaller units nearer to the sources of information, where they can "process" it in real time. The practical hallmarks of this approach are increased autonomy and a wider diversity of working styles. The *Times* writes,

> Today, it is not only Soviet-style governments that have been forced to undergo change but large corporations like I.B.M., which nearly collapsed and has had to create more entrepreneurial styles of work for its employees. With such a focus on individual initiative and the accompanying access to vast quantities of information, it is not surprising that broad ideological approaches are losing out to a case-by-case, problem-solving one (Bronner, 1999, online).

Autonomy for the individual takes the form of self-reliance in discovering needs and marshaling resources to meet them. Increasingly, it takes the form of actual self-employment and consultantship. But it also exists in terms of small groups of people, when they spontaneously form and act as a relatively independent entity, collectively solving a problem or meeting a need. Autonomy then works on at least two levels: the team itself is relatively autonomous, and so are the individuals within it. They may even be distributed over large geographical spaces, in which case they have considerable autonomy in terms of personal time and space.

In *The Distributed Mind* (1997), Kimball and Mareen Fisher discuss distributed, autonomous groups of this nature, which they label "self-directed work teams." They define them in the following way:

> [A self-directed work team is] a group of employees who have day-to-day responsibility for managing themselves and the work they do with a minimum of direct supervision. Members of self-directed teams typically

> handle job assignments, plan and schedule work, make operational decisions, and take action on problems (p. 44).

The members of self-directed work teams have high degrees of distinctive expertise. The roles in such teams are often highly differentiated, with experts in different fields bringing their skills to bear on the issue at hand. As for the lack of centralized control, people in self-directed work teams are themselves self-directed, locating needs and addressing them with a high degree of independence and initiative. This high degree of localization and lateral interaction makes a self-directed work team behave in ways that can be described as complex. Their knowledge is continuously constituted by their conversations and their iteratively modified products.

Continuously constituted knowledge is collaborative in the sense that it cannot exist without mutual interaction. For this reason, I define collaboration as "the act of collectively creating an ecosystem of knowledge in order to produce a desired product." Creating an ecosystem of knowledge requires continuous and fine-grained interaction. For example, writers, editors, and researchers need to work very closely together to mutually define needs and the precise forms in which they are met, and then work to produce the desired intermediate objects (paragraphs, word choices, library articles, etc.) leading up to the final product. The work process is highly iterative, with multiple drafts being produced which draw ever nearer to a collectively acceptable product.

This definition differs from earlier definitions of collaboration. Goodman and Abel (1987) more traditionally define collaboration as "the reciprocal exchange of information, people sharing information of some form and, thus, effecting changes in the thinking and actions of the people involved in the process" (p. 130).

Trimbur and Braun (1989) distinguish between "hierarchical" and "dialogic" forms of collaboration, pointing out that scientific collaboration is often hierarchical, relying on a stratified division of labor, while teachers typically facilitate dialogic collaboration, which regards participants as peers of equal status (p. 22). The risk, they point out, is that dialogic collaboration may not map well onto workplace forms of collaboration, a concern which has also been articulated by others (Selfe, 1992; Lay, 1992). Roschelle and Teasley (1995) offer essentially the same distinction in distinguishing between cooperation and collaboration, whereas the former is accomplished by division of labor and the latter is accomplished by the "mutual engagement of participants in a coordinated effort to solve the problem together" (see also de Graff, Gerrits, & Huysman, 1999).

The advantage of a complexity-oriented model of collaboration is that it permits differentiation of roles, unlike the dialogic model of collaboration, but does not require that those roles be stratified, as does the hierarchical model. Leadership roles can shift among members of a team over time. These roles may be distributed as team members shift periodically to inhabit their areas of greatest competence. The team members are self-directed actors in an ecosystem, not cogs in a machine. Students become "producers, gatherers, and consumers of information" (Guzdial, 1997, p. 84). Similarly, Gill (1999) uses the term "collaboration" "to refer to how unique perspectives, expertise and contributions can achieve a convergent outcome, much as an ecosystem entails the integration of varied types of participation" (p. 205). This is why I define collaboration as the act of collectively creating an ecosystem; it entails close cooperation in producing, gathering, and consuming information.

This systems-oriented definition of collaboration suggests that teachers should design classroom environments where students can work in self-directed teams to perform, as authentically as possible, the roles that knowledge workers play. This entails creating classroom environments with a high degree of locality, e.g., where students work relatively autonomously within teams which are themselves relatively autonomous. Given the complexity of such environments, management techniques can help greatly in running them. Teachers need to know how to facilitate the planning and implementation of projects, collaborative creation of documents such as reports and software, the creative generation of new ideas, peer teaching, and negotiation. This translates into classroom techniques such as the following:

- Increasing the emphasis on teamwork and project-based learning.
- Explicitly teaching students techniques of communication, negotiation, and conflict resolution.
- Shifting the teacher's role from lecturing to coaching and strategic intervention.
- Emphasizing situated knowledge through internships, apprenticeships, and classroom "businesses."

Using these techniques successfully is a pedagogical challenge. Left to themselves, student groups often substitute cooperation for collaboration, for example by trying to write the parts of a paper separately and then pasting them together, rarely with good results. One goal of teaching in complex environments, then, is to help students move from cooperation to true collaboration. This is done through coaching, through teaching students techniques of communication and

negotiation, and by creatively giving them opportunities to make decisions about their learning (Kohn, 1993, pp. 221-227).

It should be noted that an overly literal emphasis on locality can have negative effects. It can be used to justify designs which balkanize classrooms into disconnected teams. It can inhibit teachers from exercising appropriate control, such as required readings and lectures – methods which are deemphasized in an autopoietic pedagogy, but which still have a place. A successfully constructivist classroom will always be a balancing act between the centrifugal forces of localization and the centripetal forces of central control.

Topology

Theories of complex systems have a distinctive topology: they privilege networks over hierarchies. Models of complex systems focus on the multiple and lateral interactions between units, whereas hierarchical models privilege one-way and top-to-bottom interactions. Manuel Castells (1996) suggests that the network is the characteristic form of knowledge-centric organizations. According to him, the "network enterprise" is "that specific form of enterprise whose system of means is constituted by the intersection of segments of autonomous systems of goals" (p. 171). Castells' definition emphasizes the relative autonomy of the organization's components, and the fact that they interact with rather than control each other. Spatially, network enterprises are characterized by a flat hierarchy and team-oriented units, and seek to increase their "surface area" of knowledge-gathering by maximizing their contacts with suppliers and customers. They also continuously carry out training at all levels, to maximize the inflow of new knowledge (p. 164).

Network topologies overcome the isolation of groups from each other. Each group has a unique skill or function relative to its immediate environment, which could isolate it were there not lateral connections between it and other groups. In the classroom, a network topology comes into existence when students discover that they can draw on the expertise of classmates in teams outside their their own. In fact, this ability is becoming recognized as a crucial skill in knowledge organizations. In an interview with Jim Spoehr, Ted Kahn (n.d.) speaks of the importance of what he calls "know-who":

> Effective learning is as much about people finding and helping other people who can assist them in their own pursuits than it is about just finding and using information.... There's something very intuitive about finding key people who have the right kinds of ideas, talents, or resources you need at just the right time. I think this is something that master deal makers or "knowledge brokers" have developed into a fine skill (Online).

Students achieve "know-who" both by social networking and by being able to learn about other students and teams through computer networks. In Chapter 2, I will discuss my efforts to facilitate "know-who" in the classroom by using several features of the Collaboration Center, including the roster and the team rooms.

Ecology

Theories of distributed organization tend to reach for ecological or organic metaphors rather than mechanical ones. Instead of envisioning the organization as a machine, they envision it as a life form, ecosystem, or economy. Through Peter Senge's perspective on learning as a *generative* activity, we come closer to a view of knowledge as being the product of a quasi-living system of continuous interactions. This organic quality is also present in Garvin's definition of a learning

organization (1998): "A learning organization is an organization skilled at creating, acquiring, and transferring knowledge, and at *modifying its behavior* to reflect new knowledge and insights" (p. 51, emphasis added). The choice of an organic metaphor has significant implications for theories of teaching, for one interacts with an ecosystem in very different ways than one interacts with a machine. One designs, controls, and repairs a machine, but one adapts, perturbs, and maintains an ecosystem.

Feltovich, Spiro, Coulson, & Feltovich (1996) go so far as to suggest that collaborative work is pedagogically effective because the "organicistic" quality of the discourse models the equally "organicistic" properties of the processes that are the object of learning. They propose that in gaining an intuitive grasp of the way discourse works, students also gain an heuristic model which can be applied to the way other complex processes work. They write, "Group learning... may provide a special vehicle for the learning *of a new way of thinking and perceiving* – in this case, the inculcation of an organicistic worldview" (p. 39, italics in original). Along similar lines, Mitchel Resnick (1994) suggests that simulation-based learning with cellular automata will inculcate a new generation with the habits of thought required for visualizing and explaining complex, decentralized phenomena (p. 29). These hypotheses directly connect constructivism, a theory of pedagogy, with autopoiesis, a theory of organic structure.

The ecosystem metaphor offers a number of helpful advantages for working with network organizations. First, it draws attention to the preexisting conditions of a human organization, since one can never start an ecosystem from scratch but rather has to build upon what has already developed. This steers managers and teachers away from believing that they can fully specify and control the way

organizations and classrooms work, and toward a "bricolage" mindset of adapting what exists. Thomas Davenport (1997) stresses the importance of observation and description: "Given the multiplicity of information sources and uses…predicting the future is virtually impossible. For example, a company can't know who its competitors will be in the future or what information about them it will need; it makes much more sense to focus on describing the competitive information a company has *today*" (p. 32, italics in original).

Second, the ecological metaphor foregrounds the probability of continuous change. In biology, the idea of a stable equilibrium has been supplemented by the concept of *punctuated* equilibrium, where populations fluctuate considerably and often at the expense or benefit of particular subgroups. An ecological metaphor incorporates the recognition that teams will fluctuate and change, and form and disband, as circumstances warrant. Quasi-ecological metaphors of fluidity and change run deep in Richard Lanham's *The Electronic Word* (1993). Lanham envisions a college curriculum which behaves like "a continually remodeled and adaptive self-conscious work of art" where continual experimentation goes on in syllabi, hiring, and departmental structure (pp. 115-116). In my own courses, students formed and regrouped teams throughout the semester, while the syllabus, which was an online document, changed continually to reflect the status and direction of the class. The concept of a fixed syllabus is irrelevant in such an environment.

Third, it encourages a model of leadership as facilitation rather than control. The goal of teaching becomes to set broad objectives, create as rich an environment as feasible, and remove obstacles, rather than manage the specific details of work.

Teachers set parameters and expectations, and limit their role to strategic, targeted interventions. Details are left to the improvisational ingenuity of the students.

Facilitators and negotiatiors can offer rich examples of their craft. Robert Hargrove (1998) describes the role of Terje Larsen, a Norwegian mediator in the Oslo peace talks:

> Larsen declared at the first meeting when the delegates urged him to participate that he would only intervene "if it got into a fistfight." Larsen's main role took place between meetings: evoking images of aspiration, or even fear, urging people back to the table when they got discouraged, as well as making it clear to both sides that he and the other Norwegians deeply understood how they felt. During lunch and dinner conversations, he avoided taking a position on controversial items, preferring instead to encourage moments of recognition by reframing what one side said to another, so they could hear it better (p. 82).

Hargrove notes that Larsen focused on building relationships rather than achieving specific goals. Similarly, Roger Schwarz (1994) declares, "Developmental facilitators' ultimate goal is to work themselves out of a job by having the group learn to manage its process" (p. 91).

These models of facilitation set a clear target for change in teaching methodology. They displace the Socratic methodology of teaching, where the teacher is the focus of the conversation, in favor of a model where the teacher works behind the scenes to build robust student groups which can solve their own challenges with minimal teacher intervention. Harvey Wiener (1994) makes the same point when he argues that constructivism defines the teacher primarily as a "task setter"(p. 134) and "classroom manager" (p. 136).

In fact, Kenneth Bruffee's description of the teacher sounds strikingly similar to Kimball and Mareen Fisher's description of the manager. Where Bruffee (1993) asks how one teaches in an environment when neither the teacher nor knowledge itself has ultimate authority, Fisher and Fisher ask how managers manage in a nonhierarchical, nonauthoritarian setting. In both cases, the leader is seen as designing an environment rather than specifying the exact process by which activity takes place. Of teaching, Bruffee writes,

> Teachers teach for the most part indirectly, through reorganizing students socially and designing appropriate tasks. Students converse among themselves with the teacher standing by on the sidelines, for the time being mostly ignored (p. 31).

He also writes:

> To teach writing is to create conditions in which students learn to converse with one another about writing as writers do, and it is also to create conditions in which students learn to write to each other as do the members of the community of literate people (p. 73).

Bruffee discusses this facilitative style of teaching in detail, suggesting many ways that teachers can construct environments which guide students through the challenging process of working together. Chapter 2 of his book is devoted to "environmental setup" techniques that teachers can use.

For their part, Fisher and Fisher make highly similar suggestions about what they call "hands-off" leadership, where the manager's efforts go toward structuring the environment and coaching subordinates:

> The boundary manager focuses on the environment that surrounds the team…Rather than directing her primary energy to the throughput process,

> the team leader focuses more attention on boundary issues such as interface problems with other teams, customer and vendor interactions, dealing with other corporate groups, assessing competitors and market opportunities… (p. 199).

Fisher and Fisher also write, "Boundary managers don't direct work. That's not their role. While traditional managers usually work *in* the system, boundary managers work *on* the system instead" (p. 200). It can be seen that there is a high congruence between these two visions of teaching and managing.

Thinking of teachers as "boundary managers" is useful because it helps define where the teacher's responsibilities are and are not. The grounds of evaluation are not prowess at lecturing but how well the students work with the tasks they are set, and the teacher's skill at synthesizing their work and helping them see it in a larger perspective. A hands-off approach to group work, where the teacher stays out of the groups and even leaves the room is often recommended so that groups will develop their own decision-making capacity (Michaelsen, Jones, & Watson, 1993; Wiener, 1994). Hepzibah Roskelly (1994) takes Wiener's recommendations a step further by urging teachers to "make ourselves brave enough to risk the dissent that inevitably comes when democracy is in action" by relinquishing control over class discourse (p. 146). Batya Kahn and Peter Friedman (1998) have noted that with teacher guidance, self-governance works surprisingly well in the classroom (p. 169). The teacher's job is to set up the boundaries, tend to them, and intervene strategically but sparingly. What he or she does not do (or does less, at any rate) is lecture, provide answers, be the focus of the classroom, and control every moment of the students' engagement with the subject.

This is not to say that the teacher is abandoning her responsibilities, "taking it easy" – though she may feel that way for a while as she adjusts to this way of teaching. The hard work happens before and after class, as the teacher thinks about the "boundaries" of the classroom: the exercises that prompt and mediate students' interactions, feedback on group and student work, synthesis of reports from various groups, and one-on-one interactions outside of class. This kind of teacher achieves her primary impact through carefully timed and calibrated interventions, nudging students in the right direction at the right time, rather than through the broadscale technique of lecturing.

George Hillocks (1995) has elaborated a concept of "environmental teaching," and it is clearly suited to the conception of the classroom as a decentralized organization. As Hillocks defines it, environmental teaching is "teaching that creates environments to induce and support active learning of complex strategies" (p. 55). He offers an example of this kind of hands-off, discourse-oriented teaching from his days as a student teacher observing classes:

> He [Bernie McCabe, the teacher] walked to the other side of the room, seated himself in the rear corner behind his desk, and picked up what appeared to be a packet of three-by-five cards. I was puzzled. He seemed not to care about his students at all. As I was sitting down, a girl walked to a teacher's desk at the front of the room, struck a gavel on it, and called the class to order. She called on "the secretary" to read the assignment for the day. A boy stood up and read it, including pages to have been read, a brief report to have been written, and what appeared to me to be an outline of what the class was expected to know. I was astonished. Next, a different student took the floor and began asking questions about the geography, politics, and economy of Eastern Europe. Ten to 15 students had their hands up for each one. I looked furtively over to McCabe, who seemed to be paying no attention, just making notes on his little cards. The same students directed the class through several student activities, including a small-group session in which students shared folktales they had been

> collecting from relatives. During the final few minutes, the class turned to the reading that a student had assigned for the following day. McCabe met with a small group of students at his desk, his first direct intervention in the double period (pp. 2-3).

It seemed to Hillocks that McCabe wasn't doing anything. But in fact McCabe must have done a great deal of preliminary work to show his students how they were expected to behave, and to construct activities which would maximize the effectiveness of their work together. His interventions were targeted rather than broad (i.e. meeting with a small group at his desk). And in taking notes, he was likely recording observations for future interventions and evaluation.

Despite McCabe's example and decades of research on learning, transmission-oriented, lecture-based teaching is still the dominant method of education in colleges and high-schools today. Methodological change is difficult, and teachers lack support and direction for implementing constructivist alternatives (Bonk & Cunningham, 1998, p 31). Furthermore, as Gerald Graff (1987) has written, the institutions of teaching have a structural conservatism which makes true change difficult to implement. Graff observes that the field-coverage model of teaching instituted in the 1880s, where instructors specialized in genres and periods, had the effect of isolating teachers from each other. It also isolated innovative methodologies by grafting them onto the system as new courses and departments, leaving existing ones unchallenged. Real change in teaching tends to happen from the bottom up, by innovative (or new) teachers trying new things, rather than from the top down.

Another way to describe the educational system is in terms of actor network theory, in which it would be "loosely coupled" (Morrison & Goldberg, 1996, p. 126). A loosely coupled network is one in which the components are relatively

autonomous and relatively slow to be affected by the external environment or internal controls. While a loosely coupled network has the great advantages of diversity and resilience, it also makes it difficult for administrators to exert significant top-down influence or even reward progress.

Still, as educators become more aware of the situatedness of knowledge and the decentralized nature of 21st-century workplaces, it becomes increasingly difficult to justify a transmission model of education. For their part, industry educators – or knowledge managers – are increasingly aware of the high costs and relatively low yield of formal classroom training. They are more likely to turn to ways where learning is embedded in work activity itself (Balla & Andrews, 1999, p. 32). Furthermore, there is a new recognition that management, among its other duties, entails the production of learning environments. Shoshana Zuboff has documented the considerable increase in cognitive demands which comes from moving from a paper-based to a computer-based work environment. Computers produce reams of information, which workers become obligated to analyze and interpret. Productivity rises, but so does the complexity of the environment. Even apparently rote jobs such as mixing pulp in a paper mill become much more information-oriented once the factory's machines are converted to computer-controlled units. Zuboff argues that managers must pay attention to helping workers think independently, absorb new information, and use it to productively to solve problems and improve end products. Managers, then, must devise ways to help their charges learn if their organization is to keep pace:

> The informated organization is a learning organization, and one of its principal purposes is the expansion of knowledge – not knowledge for its own sake (as in academic pursuit), but knowledge that comes to reside at the core of what it means to be productive. Learning is no longer a separate activity.... Learning is not something that requires time out from being

> engaged in productive activity; learning is the heart of productive activity. To put it simply, learning is the new form of labor (p. 395).

An example of an industry learning environment is the "action learning" philosophy, described by Michael Marquardt (1997). Action learning is a methodology where facilitated teams reflect on a task at hand and develop active solutions to problems. The learning process is explicitly embedded in a problem-solving activity.

In sum, good teachers have to think in managerial terms, and good managers have to think pedagogically. The jobs of teacher and manager are becoming increasingly similar to each other, and they are both becoming more oriented to the vision of the organization as an organic entity which can be shaped but not controlled.

4. The Limitations of the Ecological Metaphor

While the ecological metaphor has considerable advantages for talking about social organizations, it does have its limitations. The main one is that biological systems have no teleology, whereas human organizations do. There is no purpose inherent in a forest or an ant colony, nor in the actions of individual actors within it, beyond that of survival and reproduction. By contrast, human organizations articulate specific goals and move purposively to achieve them. Thus the idea of design is more important in a human organization than the ecological metaphor would suggest. While managers cannot fully control the conditions of an organization, as discussed above, they nonetheless exert considerable deliberate influence over its basic parameters. Managers hire and fire, teachers create syllabi,

and so forth. Therefore *leadership* is an important issue in human organizations, whereas it is irrelevant in biological systems.

How can management theorists reconcile the purposelessness of "natural" ecological systems with the purposiveness of human actors? Organizational theorists such as Robert Hargrove, and Suzanne Kelly and Mary Ann Allison, suggest that intentionality should focus on the task environment rather than the tasks themselves. The manager, and the teacher, are charged with maintaining a healthy environment, within which work develops in characteristically organic and often unpredictable ways. Environmental strategies include cultivating sources of information, managing relationships between groups, creating a software infrastructure to facilitate communication, and coaching on an *ad hoc* basis to resolve problems. In this way, managerial intentionality is focused on maintaining *systems*, within which worker intentionality can focus on accomplishing *tasks*. A useful metaphor for this reconciliation is that of gardening: the gardener's goals and responsibilities are to create a healthy growing environment, within which the plants have their own "goals" of growth and reproduction.

The gardening metaphor acknowledges that there may be some degree of disconnect between managerial and worker intentionality. They may have different working goals, on a day-to-day basis. This disconnect is not a problem if the workers are substantially self-directed, as emphasized by Fisher and Fisher. In such a case, there is less of a need for managers to explicitly articulate goals and impose them on subordinates.

5. Family Resemblances: Complexity Theory, Constructivism, Network Pedagogy

The likeness between complexity theory and constructivist pedagogy has a long history. Philosophers and educators have frequently rejected the model of education as a "passive acquisition or absorption of an established (and often rigidly defined) body of information" (Koschmann, 1996, p. 5). The common denominator among them is the recognition that learning is an active, organic process, and that knowledge cannot be separated from the context in which it arises. At the beginning of the 20th century, John Dewey's *Democracy and Education* (1916) carried out a withering attack upon educators who view learning as a passive, teacher-driven process. He argues against their emphasis on arranging and delivering information, seeing it as the "freezing" of knowledge into an abstracted and dead form. Against their methodology of transmitting carefully arranged facts, he proposes that education's essence is to train "vital energy seeking opportunity for effective exercise":

> The conception that the mind consists of what it has been taught, and that the importance of what has been taught consists in its availability for further teaching, reflects the pedagogue's view of life.... It exaggerates beyond reason the possibilities of consciously formulated and used methods and underestimates the rôle of vital, unconscious attitudes. It insists upon the old, the past, and passes lightly over the operation of the genuinely novel and unforeseeable. It takes, in brief, everything educational into account save its essence, – vital energy seeking opportunity for effective exercise (pp. 71-72).

For Dewey, learning is an active, dialectical, dialogical, situated, ongoing process, occurring in virtually all contexts of human activity rather than just in the

schoolroom. He did not theorize learning in a systems-oriented framework, but his ideas about learning map easily onto one.

Dewey's conception of learning as a process has become established in composition pedagogy. Maxine Hairston (1982) has proposed that rhetoric is experiencing a "Copernican shift" from product-based pedagogies to process-based ones, emphasizing discourse and negotiation rather than arrangement and delivery. The shift to process models of discourse has influenced writing texts such as Linda Flower's *Problem-Solving Strategies for Writing* (1993), which view writing as a socially grounded, self-reflexive activity.

In the same vein, Kenneth Bruffee draws from pragmatist and antifoundational philosophies in his book *Collaborative Learning* (1993) to argue that it is teachers' jobs to acculturate their students into new discourses by having them learn to "speak" them to each other through discussion, debate, and open-ended research. Since, Bruffee argues, knowledge is not a fixed and timeless absolute but rather a human institution subject to changing practices, the goal of education should be to encourage students to understand a field's discourse *as* discourse, recognizing both its power and contingency. Though Bruffee does not use the terminology of autopoiesis, his pedagogy is grounded in an organic epistemology of knowledge. For Bruffee, knowledge is a quasi-living entity, shaped in continuous discourse. This recognition enables us to map Bruffee's language onto the theories of complexity outlined above.

In their turn, complexity theory and constructivism are also compatible with methodologies of teaching in networked environments. Fred Kemp (1993) coined the phrase "network theory" to describe the joining of constructivist pedagogy and

networked computing. At the University of Texas at Austin, a generation of drill-and-practice programs was swept away in the late 1980s by the influx of constructivist theory. Kemp recalls his moment of insight and conversion to the new methodology:

> [At the 1987 CCTE conference] their group presented a session on collaborative instruction and the pedagogy of Kenneth Bruffee. Although Butler remembers being worried by all the nudging and whispering Paul and I were doing in the audience, we were actually responding to what was being said with a great deal of excitement. Here was the classroom theory to fit the network, and it represented, to us at least, a considerable shift from what we had always thought of as the proper way to teach writing (Kemp, 1996).

This conversion to "network theory" was shared among Fred Kemp and Jerry Bump, John Slatin, Locke Carter, Kay Halasek, Lester Faigley, and others at the University of Texas at Austin. They recognized that computer networks could facilitate the kind of open-ended and decentralized communication that goes on in an organically evolving human organization. Historically, computer networks have both mirrored and modeled human organizations. That is to say, while the software inscribes a model *of* the organization as it is used over time and accumulates documents and other data, it also offers a model *for* the organization. As Sherry Turkle (1995) notes, software offers new metaphors for envisioning institutions and practices. It can be, to use Turkle's phrase, an "object-to-think-with" which influences an organization's dynamics and structure (p. 14).

6. Network Pedagogy in the Technological Classroom

As objects to think with, computers draw students into them, to work on projects, solve problems, and contribute to discourses facilitated by and preserved

in virtual spaces. Class activity shifts away from lectures and toward projects, which suits constructivism's "problem-based" pedagogy. This makes the computer classroom a space of improvisation – or, as Lave and Wenger (1991) describe it, the site of a "learning" (as opposed to "teaching") curriculum:

> A learning curriculum consists of situated opportunities.... for the improvisational development of new practice. A learning curriculum is a field of learning resources in everyday practice *viewed from the perspective of learners*. A teaching curriculum, by contrast, is constructed for the instruction of newcomers. When a teaching curriculum supplies – and thereby limits – structuring resources for learning, the meaning of what is learned...is mediated through an instructor's participation, by an external view of what knowing is about (p. 97, emphasis in original).

The computer classroom, with its attendant software, constitutes a "field of learning resources." Of course, not all computer classrooms are correctly designed to be sites of learning curricula. Some of them are clearly intended to be sites of *teaching* curricula: the computers are arranged in rows facing the front of the room, with a teacher's lectern as the designated focus of attention. This is the worst possible way to set up a computer classroom. Teachers soon discover that the students' first instincts are to engage the computer, begin working on projects, or start working, either face-to-face or virtually, with other students. If teachers wish to reassert a "teaching curriculum," they have to impose unusually heavy discipline, or even force.[2] Charles Moran (1998) writes,

> [T]eachers new to our computer classrooms often find the fact that students are *not* looking at them to be profoundly disturbing. To become themselves

---

[2] There is a company that sells networking software which allows the teacher remotely to view any computer at will, and freeze all the students' computers at once. I used such a system while teaching at the Texas Department of Health, and I found that students usually resented such heavy-handed control.

> the center of their students' attention, they often break their students away from the computers, pulling them into a circle as far from the infernal machines as they can get (Online, italics in original).

More enlightened computer classrooms embody, in architectural form, Bruffee's dictum that the teacher should "stand by on the sidelines." In the computer classrooms run by the Computer Writing and Research Lab (CWRL) at the University of Texas, the physical layout of the computers directs students' attention away from the center or front of the room to its periphery. The reconfiguration of space parallels an equally important reconfiguration of power. In this reconfiguration, the teacher moves from being a centralized authority vested with absolute knowledge to being a manager of discourse.

This architectural and epistemological reconfiguration of learning spaces facilitates engagement with problems whose solutions are not neatly defined. In "Computer-Supported Problem-Based Learning: A Principled Approach to the Use of Computers in Collaborative Learning" (1996), Koschmann, Kelson, Feltovich, and Barrows suggest that education must be designed to teach students how to work in "ill-defined domains." An ill-defined domain exists where "defining the problem requires more information than is initially available – the nature of the problem unfolds over time; there is no single, right way to get that information; as new information is obtained, the problem changes; decisions must be made in the absence of definitive knowledge; and there may never be certainty about having made the right decision" (p. 87). Of course, working life consists of ill-defined domains, and Koschmann et al. criticize formal education for having studiously avoided the messy contingencies associated with them.

One way to introduce students to ill-defined domains is to present them with problems to be solved, as opposed to abstract discussions of principles; hence the term "problem-based learning" (see also Cameron, Barrows, & Crooks, 1999, p. 86; Duffy, Dueber, & Hawley, 1998, p. 59). For Koschmann, who is a medical educator, worthy problems include simulations of difficult medical cases, sometimes presented in multimedia format, sometimes in labs with actors playing the role of patients. As students progress to internships, the problems become increasingly authentic as they work with real patients under the guidance of more experienced doctors.

For Koschmann, computers are a crucial component of education in ill-defined domains. They can be used to satisfy the following needs, among others:

- the need for an collection of authentic cases with a mechanism for selecting appropriate cases.
- the need for a retrievable record of the group's deliberations.
- the need for multiple viewpoints uncontaminated and uninhibited by group effects.
- the need for students to be able to share information outside of meetings.
- the need to readily access learning resources.
- the need for students to be able to index their notes in a way that will facilitate later retrieval (paraphrased from p. 104).

These needs are often solvable with the aid of network-based solutions. But Koschmann goes further than simply proposing specific networked solutions to particular needs. He argues that computing both inspires and supports a

theoretically rich pedagogy, one which is designed to acculturate students to working in complex, ill-defined, autopoietic environments. His pedagogy is founded on six principles of learning:

> Principle of Multiplicity: Knowledge is complex, dynamic, context-sensitive, and interactively related; instruction should promote multiple perspectives, representations, and strategies.
>
> Principle of Activeness: Learning is an active process, requiring mental construction on the part of the learner; instruction should foster cognitive initiative and effort after meaning.
>
> Principle of Accomodation and Adaptation: Learning is a process of accomodation and adaptation; instruction should stimulate ongoing appraisal, incorporation and/or modification of the learner's understanding.
>
> Principle of Authenticity: Learning is sensitive to perspective, goals, and context, that is, the learner's orientation, goals, and experiences in the learning process determine the nature and usability of what is learned; instruction, therefore, should provide for engagement in the types of activities that are required and valued in the real world.
>
> Principle of Articulation: Learning is enhanced by articulation, abstraction, and commitment on the part of the learner; instruction should provide opportunities for learners to articulate their newly acquired knowledge.
>
> Principle of Termlessness: Learning of rich material is termless; instruction should instill a sense of tentativeness with regard to knowing, a realization that understanding of complex material is never completed, only enriched, and a lifelong commitment to advancing one's knowledge.

Examples of how computers support these principles can be seen in how Koschmann uses the Collaborative Learning Laboratory (CLL) developed at his university. The CLL augments, rather than replaces, the interactions typically seen in small-group work. For example, realistic case presentations (high-fidelity, multimedia, uninterpreted representations of clinical data) are an example of the

principle of authenticity; they are hypertextual collections of information about a case, which the student can browse at his or her own initiative. They are drawn from real cases, and the undigested nature of the material resembles how the student would encounter the case in real practice. Similarly, the principle of articulation is facilitated by the chat and bulletin board features of the CLL.

Koschmann's principles can be seen in action in many other pedagogical contexts. The principle of accommodation can be seen in Neuwirth and Wojahn's discussion of how a networked collaborative writing program, PREP Editor, solved some of the limitations of face-to-face discussion between students commenting on peer drafts (1996). In storing students' annotations of their peers' paper drafts, it encouraged activeness, adaptation, and articulation. The latter is especially seen in contrast with face-to-face discussions, whose ephemerality caused them to lose an important part of their value:

> An exploratory study suggests that students who talk face-to-face about their writing tend to classify remarks into two categories: those they agree with and those they do not. Students tend to jot down only those remarks with which they agree and, as they revise their drafts, they tend to ignore those points they dismissed. Because face-to-face interaction is verbal and ephemeral, it is difficult for participants to reflect on their own interpretations of the interaction, even when they try to take notes (p. 161).

PREP Editor was valuable in that it textualized commentary, preserving it for reflection and further dialogue. Similarly, John Slatin (1992) argues that electronic chat environments support what he calls "recursion," the process of reflecting upon and incorporating earlier discussions into later ones (Koschmann might call this an example of both activeness and termlessness). Slatin found that the students in his courses which used real-time chat kept more mental resources "in play" than did students in his traditionally taught courses. The availability of

chat session transcripts for consultation allowed students to refer more frequently to more authors in relationship to each other, constructing a more densely nuanced understanding of resonances between them. Slatin suggests that this denser understanding enabled students to think about their textbooks metacritically, offering comments about the order in which poems were arranged and studied. This move from the meaning of individual poems to their contexts and metacontexts exemplifies the principles of multiplicity and authenticity (that is, the students began to talk about literature in ways more nearly resembling the ways that authors and editors do.)

Koschmann's principles can also be seen in discussions of industry learning, such as Wanda Orlikowski's study of a tech support group at "Zeta," a pseudonym for a major software manufacturer (1996). When the department migrated to a Lotus Notes database of tech support clients and solutions, it saw dramatic gains in its ability to solve problems. This happened because technicians could draw on others' expertise more rapidly through the Notes system. Furthermore, the solutions themselves became better-documented as technicians discovered that good notes not only helped other technicians more, it also accelerated their own learning (Koschmann's principle of articulation.) In addition, the database became a source of case studies for training new employees (the principle of authenticity), and its realism brought new hires up to speed in five to six weeks instead of the eight it had taken previously.

7. Managing Complex Learning Environments with Software

Koschmann's work offers a well-theorized pedagogical basis for using computer networks in the classroom. However, it says relatively little about how

teachers can manage the complexity of the interactions which emerge. There are diverse projects in various stages of completion, with high message traffic between all participants. Various individuals and teams need coaching at various times, which requires the teacher to develop ways of becoming aware of when coaching is wanted. Contingency is the rule, where problems are ever arising and being solved with creative interventions. In such an environment, how does the teacher keep up? For that matter, how do the students keep up?

Software for online teamwork is intended to address just this problem. It is often called "groupware" to call attention to the fact that it creates an interactive, multi-user environment for the purpose of coordinating communication and collaboration within groups. It is created with the *context* of work in mind (Bock & Marca, 1995, p. 9) and can become a context in its own right (Goodman & Abel, 1987). It is intended to create an encompassing environment for collaborative work, instead of simply supporting narrowly defined activities. It typically addresses well-defined needs such as messaging, calendaring, scheduling, polling, and document management, but it also addresses less well-defined needs such as casual social interaction, awareness of peripheral activity, negotiation, conflict resolution, and accumulation of intellectual capital. If both sets of needs are satisfied, groupware can robustly support an ecosystem of knowledge.

Groupware presents itself most immediately as technology, but this characterization is misleading, because it portrays groupware as something that exists outside human relationships and is grafted onto them. Furthermore, it carries unfortunate baggage: the assumption that groupware is *only* technology and belongs to the realm of information technology departments and programmers. These are usually not the people who should be solely responsible for designing

and implementing groupware solutions. Groupware solutions are part of a larger class of communicative and procedural solutions, which software designers are rarely trained to analyze and solve. Winograd and Flores (1987) warn that software appears to be an autonomous entity but is really a form of intermediation between one human being and another (p. 123). In working with the software, users confront not a machine, a fundamental Other, but a set of human decisions made tangible. Groupware is a blank slate upon which designers and users inscribe their ideas about work and relationships.

It is crucial, then, to view groupware not simply as a technology but as one among many communicative domains where groups carry out their work. It is useful to view a groupware program as a "document," not in the commonly accepted sense of the word but in the expanded sense put forward by John Seeley Brown and Paul Duguid in their essay, "The Social Life of Documents" (1995). They take issue with the traditional conception of documents as (a) the conduits by which meaning travels from one party to another, and (b) as the end product of discourse. In place of these ideas, they suggest that documents are sites where meaning is negotiated within communities:

> Shared documents within communities are in many ways simply the grounds for a fight, merely the pre-text for agreement. Providing a shared context for constructing meaning, documents are the beginning rather than the end of the process of negotiation (Online).

Seen in this light, a groupware program is a "document" in two senses. First, it is a space where meaning is constructed, most typically in textual form. For example, it may include a bulletin-board style forum, where postings are literally pre-texts for continuing discussion. Another example is an online syllabus, which changes frequently as the teacher and students negotiate upcoming

assignments and deadlines. Still another example is the contents of "team rooms," where team members discuss projects and post deliverables, which themselves trigger further discussion.

Second, groupware potentially produces information about human activity, such as login data and usage data. This information can be read as a document which is "about" the team or classroom's working styles. Shoshana Zuboff (1988) has coined the word "informating" to explain the impact of this kind of information. Software plants an automated record, the paper trail as it were, of actions, transactions, and communications within the organization. A supermarket, for example, will generate many thousands of items of information each day, of items received and bought, cashier checkins and checkouts, messages from one employee to another, and so forth. Computers automate activity, creating new information in the process:

> The action of a machine is entirely invested in its object, the product. Information technology, on the other hand, introduces an additional dimension of reflexivity: it makes its contribution to the product, but it also reflects back on its activities and on the system of activities to which it is related. Information technology not only produces action but also produces a voice that symbolically renders events, objects, and processes so that they become visible, knowable, and shareable in a new way (p. 9).

When an organization places an electronic text at its core, it has, in Zuboff's term, *informated* (as opposed to merely automated) its activity. The electronic text is a text like any other, and it becomes subject to interpretation. It is consulted for debugging problems, evaluating employees, making decisions on future action, and the like. It has the effect not only of rationalizing decisions, but also of making activity visible to an unprecedented degree. This visibility has profound and far-reaching consequences. Zuboff outlines numerous examples of how the visibility

of the electronic text reconfigures manager-employee relationships, changes workers' attitudes toward the workplace, affects union negotiations, and disempowers some employees while empowering others.

The informating capabilities of groupware mean that the software environment literally becomes a document about the group's composition and evolution. Teachers may study it to learn about the current site of the class and the student teams in it, and it may become a focus of self-reflective discussion within the class. Collected data on logins and other actions may be analyzed for insight into team and group dynamics, as has been seen in many papers analyzing chat transcripts. In a way, then, it can be said that a classroom "writes" a documentary representation of itself, over time, within the electronic space of a groupware program's database.

The availability of these two kinds of documents (the one written cumulatively by humans as they use the software, the other written cumulatively by the software) makes the electronic environment a rich space for the continuous construction of meaning. With autonomously acting students and student teams, multilateral interactions, and the ability to change and upgrade the environment to meet new needs, groupware offers considerable scope for organic and autopoietic behavior. The groupware helps to ground an ecology of knowledge, within which students can practice the roles and techniques of workers in knowledge industries.

My own software, the Collaboration Center, became an "object-to-think-with" for the English 309 composition course I taught at the University of Texas at Austin between 1997 and 1999. I wrote the software intending to facilitate multilateral communication in a decentralized learning environment; in this respect

it became a model *for* classroom discourse. In its day-to-day use, the software became a model *of* the discourse, as it became the ground of student logins, conversations, team activities, file postings, surveys, attendance and assignment records, and email messages. As the semesters went by, my students and I discovered new uses for existing functionalities, and I refined and extended the software to meet student needs and create new capabilities. In this way the software's modeling *of* and modeling *for* class discourse became ever more closely intertwined. Indeed, the story of the software's development is the story of an growing ecology of knowledge. In Chapter 2, I will discuss how my course and my groupware coevolved to create an increasingly rich learning environment.

Chapter 2: The Collaboration Center

1. Introduction

One of the best ways to teach students how to work in information ecologies is to create one in the classroom. Student teamwork can be based on a model of decentralized, self-organizing, and organic team processes, with teacher/managers charged with constructing an environment (both physical and virtual) to support them. Effective classroom and team management takes the responsibility for developing communicative, procedural, and technological structures to support collective dialogue and collaborative work.

This vision of education goes beyond the model of teacher-as-coach. Here, the teacher is not only a coach but an environment designer and manager as well, exerting primary influence on the total environment in which students work. Michaelsen & Black (1994) stress that a critical component of this influence is empowerment, that is, giving students the means, the ability, and the opportunity to form robust teams. Much of this empowerment can happen in the physical confines of the classroom, by the teacher's skillful management of assignments, discussion groups, evaluation protocols, and so forth (Bruffee, 1993; Ede & Lunsford, 1985; Elbow, 1973; Gere, 1987; Hairston, 1982; Hillocks, 1995; Johnson et al 1984; Syverson, 1999). However, in the last decade, research has gone beyond those physical confines to explore how software expands the ability of teachers to set up environments for robust multilateral communication (Bryan et. al., 1997; Faigley, 1992; Forman, 1992; Hartman et. al., 1991; Kemp, 1993; Neuwirth & Wojahn, 1996).

This body of work has focused on the possibilities of *communication*, with reference to tools such as email, bulletin boards, and synchronous chat. Its goal has usually been to investigate the dynamics of interaction between participants in synchronous and asynchronous online verbal exchanges. Researchers have studied the dynamics of online communication by analyzing variables such as the distribution of message traffic over time and over differentials of gender, race, sexuality, and power (Kiesler, Siegel, & McGuire, 1984; Regan, 1993; Selfe, 1992; Selfe & Selfe, 1994; Slatin, 1992; Sproull & Kiesler, 1991; Warshauer, 1995; Warschauer, 1999; Wolfe, 1999, 2000). They have closely read the content of the communication by investigating how various actors address each other with respect to variables such as degree of formality, verbal tone, persons to whom messages are addressed, and the length of messages.

The work I will describe in this chapter differs in that it focuses principally on the possibilities of *management*. By the word "management," I am referring to what I described as "boundary management" in Chapter 1: the focusing of managerial effort on the classroom environment as a whole, instead of dictating fine-grained goals and actions. Boundary management assumes the relative locality (that is, autonomy) of both individuals and teams, sets broad targets and expectations, and provides an environment rich with resources.

The field of Computer-Supported Collaborative Learning (CSCL) has begun to explore how software can facilitate classroom management (Cox, 1998; Guzdial et al., 1999; Kahn & Friedman, 1997; Koschmann et.al.,1996; Scardamalia & Bereiter, 1996; Roschelle & Teasley, 1999). However, little has been written about how students react to the presence of computerized (that is, informated) activity tracking in online environments. Even less has been written about how

software can directly facilitate the project work of student teams. These deficits may exist because, until very recently, the available tools were focused on communication rather than management. As Anita Fjuk (1995) puts it, communicative tools such as chat environments permit inter*action* but not inter*dependency*. They are not well-suited for management.

As I will describe, the Collaboration Center was designed to facilitate management with a rich array of integrated tools, including bulletin boards, real-time chat, team rooms, quizzes and surveys, file libraries, syllabus generators, appointment schedulers, and storage of assignment and attendance records. These tools facilitate the multilateral discourse characteristic of information ecologies, and enrich the environment by affording new kinds of interaction, both inside and outside the classroom.

2. Relational Databases and Classroom Management

What made this new focus on management possible was my use of a powerful new Web programming language, ColdFusion (created by Allaire in 1997). ColdFusion's strength relative to other Web programming languages such as perl is its ability to manipulate relational databases. As the name suggests, a relational database is about *relationships* as well as information storage. For example, one table in a relational database might hold student names, and another file might hold courses, with metadata indicating that the two files are linked together in a one-to-many relationship (that is, one course can have many students in it). The power of relational databases is that one can easily use them to model the formal relationships in human communities. In the case of teaching, these

would be courses, classes, teams, shared deliverables, actions, evaluations, and so forth.

Relational databases cannot, of course, model informal relationships, such as classroom friendships, cliques, technical assistance from neighbors, serendipitous encounters, telling of war stories, and so on. Informal relationships exert great influence on learning (Brown & Duguid, 2000). They remain generally outside the purview of what can be represented by machines, although researchers in Computer-Supported Collaborative Work (CSCW) have sought to develop substitutes or approximations for the normal face-to-face contact that happens in a physical space (Bellotti & Bly, 1996; Isaacs, Tang, & Morris, 1996; Nakanishi, Yoshida, Nishimura, & Ishida, 1996; Tollmar, Sandor & Schömer, 1996). In this chapter, I focus on how I used relational database technology to represent formal relationships, and discuss the social practices which took place within and around those representations. In the next chapter, I will discuss the limitations of such relationships, and suggest how creative design and pedagogy can work around them.

Since relational databases enable the modeling of at least formal human relationships, they lend themselves naturally to supporting some managerial practices. ColdFusion's relational database capabilities allowed me to write code which stored a great deal of information on the status and actions of individual users, and then later to use that information in new ways. For example, I was able to write code which reported, on demand, the frequency of actions performed by users, such as logins, postings to team areas, emails to teams, and so on.

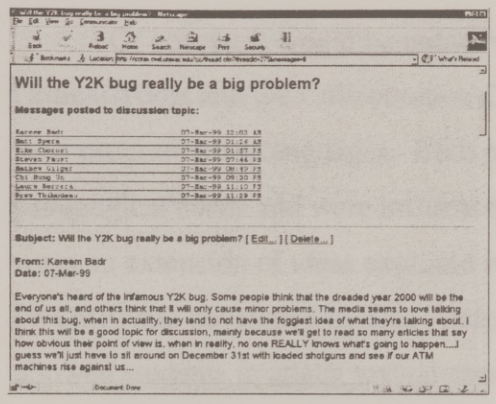
The influence of relational database technology becomes more evident when one compares two of the software projects which came out of the CWRL in 1999, the Collaboration Center and Critical Tools. Critical Tools is a set of six modules:

- Annotator (annotate passages of online texts)
- BiblioFile (build bibliographies with MLA-style citations)
- Discussion Forum (engage in asynchronous on-line conversations)
- Interactive Archives (organize materials into hypertextual libraries)
- AddLink (contribute and arrange course-related Web links)
- Class Contact (generate a class roster)

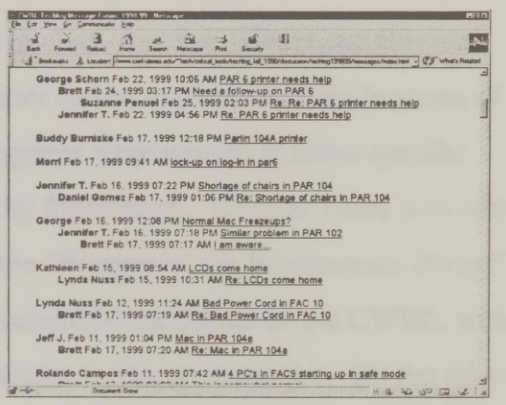
The Collaboration Center and Critical Tools overlap in offering communicative tools such as discussion forums, functions for archiving and linking texts, and mechanisms for contacting groups by email. However, apart from the Class Contact module, Critical Tools offers no facilities for class management, such as meeting or appointment schedulers, attendance/assignment trackers, syllabus generators, quizzes or surveys, and team rooms. It also has no analogue to the Collaboration Center's "Latest Events" module, which reports recent user activity.

Communication tools are Critical Tools's forte. Its implementations of communicative tools are generally more richly designed than the Collaboration Center's. For example, the Collaboration Center's forums are strictly linear collections of messages, with no visual way to signify that particular messages are responses to particular other messages. By contrast, Critical Tools indents

messages to signify sequences of topics and responses. Comparison screenshots of the two make this clear:



The Collaboration Center



Critical Tools

Figure 3: Asynchronous forums in the Collaboration Center and Critical Tools.

The Critical Tools team has thought more deeply about interfaces for textual production and interaction than I did while building the Collaboration Center. On the other hand, apart from their Class Contact module, they have spent relatively little time designing tools for coordinating classroom activity. Where Critical Tools is fundamentally about text and ways to talk about text, the Collaboration Center is fundamentally about management and coordination.

One reason for this, I believe, is that Critical Tools's programming language, perl, offers only weak database support. perl programmers can only use text files ("flat files," in database parlance) for storing data. Cross-correlating information between separate files is a complex, code-intensive undertaking. By contrast, ColdFusion enables such correlations to be made in just a few lines of code. It is not surprising, then, that perl programmers would choose to solve

problems which are focused on the manipulation of text rather than data, since that is where perl's strengths lie.

Of course, it would be reductive to claim that the differences between Critical Tools and the Collaboration Center can be explained solely in terms of their programming languages. Both programs were written to serve specific pedagogical needs and were influenced by forerunners. Critical Tools is in some ways an extension of ideas explored in the "Women of the Renaissance Period" (WORP) site developed by Dan Anderson and Morri Safran of the CWRL, which allowed students to attach textual annotations to literary passages. On the other hand, the Collaboration Center was influenced by groupware packages such as Collaborative Virtual Workspace (CVW) and Lotus Notes, which were targeted at facilitating the management of teams. However, it is possible to suggest that the programs were at least influenced by the relative strengths and possibilities of their respective programming languages.

Relational database technology raises newly pressing issues of management, oversight, and activity tracking in the classroom. These issues have long been recognized and discussed in other social contexts (Agre, 1997; Rawlins, 1998; Ullman, 1997; Weizenbaum, 1976; Zuboff, 1988), but it is only very recently that relational database client/server technology has been easily and cheaply available in a teaching context. This new availability has brought issues which have long been problematic in the workplace, and in the public space of politics, into the classroom. For example, with the availability of a robust database, it would have been easy for me to write code that recorded virtually everything a student did from the moment of login, down to the length of time spent looking at particular pages. This is extremely fine-grained, and some would say personal,

information. Should data like this be available to the teacher, and if it is, how should it be used? It is easy to write functions which would represent the state of a group project from start to finish, allowing for inspection by teachers, students, and even parents and school administrators. Logins, team documents, collective conversations, and more can be made visible to anyone the programmers allow. This potentially gives all of the actors on the educational scene unprecedented access to each other's work and activity.

In short, relational databases in client/server networks facilitate new kinds of relationships, particularly when it comes to overseeing and managing work. They go beyond facilitating isolated episodes of communication to facilitating a total ecology of communication. As I noted in Chapter 1, with reference to Castell's characterization of the "network society," the decentralized structure of computer networks bears a topological similarity to the structure of human organizations. To use Hugh Ryan's terms (1995), the "team metaphor" and networked computing are easy to align because they have the same topology:

> The team metaphor is able to take unique advantage of the recent computing technology we have been installing. Specifically, teamware can take unique advantage of the capabilities we find in client-server-based solutions (p. 480).

Ryan's conjunction of teamwork and networked computing suggests that groupware ("teamware," to use Ryan's word) has an implicit theory of social life built into it. Artifacts embody cognition. For example, a map sums up a large number of observations and calculations, and embodies human knowledge about a territory in a concretely physical artifact (Hutchins, 1995). Similarly, software can embody knowledge – or assumptions – about human interaction in teams: the structure of leadership, the needs of team members, the information individuals feel

comfortable making public, and so on. This embodiment is constituted of the cumulative decisions made by designers about design and functionality: how functions are laid out on the screen, what gets routed to where, and who gets to see what, and when. The decisions made by the designers of software have a substantial impact on how the actors on the classroom scene interact with each other. This impact increases all the more with the advent of powerful tools for modeling and representing human relationships.

3. Integrating Groupware With Pedagogy

The fact that software can facilitate classroom relationships does not mean that installing it will cause those relationships to come into being. New paradigms for teaching are required because networked computing opens up new possibilities for social relationships; its distributed, antihierarchical topology affords a thoroughgoing revision of traditional configurations of time, space, and power.

That revision is difficult to achieve. When networked computing is introduced to an environment which lacks robust theories of multilateral relationships, it can trigger a paradigm crisis (Kuhn, 1970). As networked technologies are introduced and used, their incompatibility with a traditional paradigm of transmission-based learning becomes increasingly evident. Once the initial obstacle of the software/hardware learning curve is surmounted, then the *real* problems begin to appear. A paradox arises: increased comfort with the technology eventually creates increased *dis*comfort with how it integrates into long-held theories of teaching and learning. This runs counter to the intuitive belief that teachers should get more comfortable with technology over time.

Randy Bass, a professor tenured at Georgetown in 1998 in recognition of his teaching in electronic environments, describes his experience with this paradigm crisis when he started teaching in a computer classroom in 1995:

> That semester, many things went right, but many more things went wrong.... I have come to refer to that semester simply as "the Fall" and tend to divide my whole career into two parts in relation to it – Before the Fall and After it. This is in many ways the central crisis point of my career. It was a moment when I felt as though I had lost all grounding as a teacher. I was completely surprised by the extent to which the integration of technology into my teaching changed the nature of my work and the nature of my interaction with students (Bass, 1999, online).

Bass gradually evolved a new teaching methodology over time, but such development is not inevitable. Cynthia Selfe (1992b) alludes to the powerful inertial forces which can shoehorn new tools into a weak and partial support of older, more established modes of education:

> In light of this realization, we can understand why so many school systems set up computer-supported writing classrooms that mirror traditional classrooms, even though virtual learning spaces can differ radically and productively from such spaces; why many teachers insist on lecturing within computer-supported classrooms, even though such an activity is difficult and obviously ineffective in such classrooms.... (p. 30).

Many accounts of ineffective or failed implementations of network technology have been presented in the literature (Forman, 1994; Graves & Haller, 1994; Hara & Kling, 1999; Warschauer, 1999). A 1999 Department of Education study found that only one out of five teachers felt comfortable using technology in the classroom and that schools spend only 5% of their technology dollars on teacher training (Llanos, 1999).

The failures often happen because the integration of networked computing into pedagogy is not well-managed. As Feenberg (1999) notes, educators sometimes suppose that simple familiarization with computing tools is sufficient to support good teaching with them. This approach – derisively dismissed as the "drive-by" approach by technology-literate educators (AAUW, 2000, online) – serves to familiarize teachers with technology, but does little to acculturate them to new paradigms and practices.

4. The Evolution and Content of English 309M

How, then, to develop a new paradigm of teaching in networked environments and apply it effectively to classroom practice? Like Bass, I came to a new grounding only after years of trial and error. When I started teaching rhetoric and composition courses in a computer classroom in 1995, I knew how to use the computers, in the sense that I knew how to do email, how to explore Web sites, and how to participate in chat sessions. But I had only a vague idea of how to integrate these tools into my teaching. Some uses were obvious: put my syllabus and other class information on a Web site, give my students access to me and each other via email, and have some class discussions through the chat software. But these changed my teaching style only peripherally, because I was using technology only to make my existing practices more efficient. Fundamentally, my courses consisted of lectures, classroom discussions, and individually assigned projects. As time went on, I grew increasingly uncomfortable with the disjunction between these traditional modes of teaching and the radical reconfiguration of relationships that networked computing both afforded and invited. For example, it felt increasingly awkward to lecture in a room so full of tools ready-to-hand for students to use. Furthermore, my course became overloaded with material and activities as I tried to

hold on to the old practices while also adding new ones. If one schedules class sessions to teach Web authoring, as I did for several semesters, then something else is *not* going to happen during that time. Teachers always have to make tradeoffs with class time, but the sheer newness of information technology enhanced the difficulty.

I would have remained frozen at the "discomfort" stage indefinitely, hemmed in by the technology's ill-defined relationship to long-established teaching practices, had I not encountered constructivist theory and been inspired by it while building the Collaboration Center. Constructivist theory helped me envision scenarios for situated, team-based learning, and new topologies of classroom interaction.

My first encounter with constructivist pedagogy came through Learning Record evaluation, which is a refinement and expansion of the idea of portfolio evaluation. The Learning Record seeks to put process on an equal level with product in student work. The essential difference between process and product is that whereas *process* is inextricably situated with relationship to meanings, purposes, relationships, and commitments, *products* are alienable from the conditions of their production. This alienation allows – indeed, requires – teachers to discount the unique conditions in which each learner negotiates with a body of knowledge. This alienation is the ground of traditional letter-based grading methods, which have been criticized as artificial and reductive (Kohn, 1993; Pirsig, 1984; Schank, 1994). Learning Record evaluation emphasizes narration and community in order to situate learning in a social and temporal context (Syverson, 1999, online). Students are guided through constructing coherent stories of their educational growth, and their stories are placed into a community context when

read by teachers, peers, and ouside observers. It is not merely a different method of evaluating, but a new paradigm of teaching altogether. It reconfigures the nature of student work, transforms the relationship between the teacher and the student, and expands the sphere of expertise and evaluation into the community beyond the classroom. It is readily amenable to a view of the classroom as an ecology of knowledge (Syverson, 1999).

Learning Record theory prepared me to envision software structures which facilitated information ecologies. For example, I created a Knowledge Base module where students could post stories and tips, and team rooms where student teams could post their deliverables and discuss projects at hand. I created an online diary space where students could store their "Observations," that is, ongoing notes on their learning experiences. (This later had to be changed, as I will discuss below.) I focused on accountability by putting up modules that displayed to students up-to-date records of their attendance and assignments, and by opening the Collaboration Center up to guests on a limited basis, I implicitly invited a larger community to view what was happening in my class. Thus Learning Record theory had a substantial shaping influence on the development of the Collaboration Center.

In two semesters, Spring '98 and Fall '99, I completely redesigned the course each time. Most of the changes concerned the readings, the pacing of the projects, the assignments, the scheduling of conferences, and the various ways I used the Collaboration Center in the course. By Spring '99, the course had stabilized into a workshop-, team-, and discourse-oriented configuration, with the Collaboration Center playing a major role in its information ecology. The course was about the process of collaboration itself, rather than any specific subject

matter; students had considerable latitude in the topics of their projects. The key assignments were as follows.

Initial Team Projects. To support a networked classroom topology, in the sense that students knew who their peers were and what they could do, I started each semester off with a series of three short (two- or three-day) team assignments. Students were cycled into a new team each round, enabling them to get to know roughly half their classmates before launching into the major projects. I also built a roster to help students present themselves to the rest of the class. This tool will be described in more detail below.

Research Papers. These were exercises in historiography and collaborative writing in paper documents. Student teams were asked to choose a topic, a particular magazine, and a three-year time period 50 or more years ago, and write a paper discussing how the periodical dealt with that topic during that period.

Knowledge Workshops. The purpose of these was to encourage multilateral teaching, where expertise was distributed across the class instead of being concentrated in the instructor. Students were asked to poll their peers (using the Collaboration Center) to determine a topic of interest to a significant minority of the class. Working in teams of two or three, students designed a complete lesson and taught it to the class. After the workshop, the students put up a follow-up poll to evaluate its success. Projects have included tutorials on using Dreamweaver, Flash, Photoshop, good communication techniques, good web design techniques, computer hardware, and newsgroups.

Web Debates. In the Web Debate assignment, students collaboratively created a Web site on a controversial topic. Teams of students were asked to research different positions and build branches of a Web site to put them in creative juxtaposition. The object was to foster the development of multiple and contrasted viewpoints on a subject, emphasizing multiplicity and difference instead of consensus. Teams were supported by the Collaboration Center's team rooms. Topics included cloning, genetic engineering, Y2K, college athletics, home schooling, online gambling, and the war on drugs.

Web Final Projects. Students were required to create their own Web site on a topic of their choice. The assigned size was between 20 and 30 nodes, which was sufficiently large to require careful organization and good interface design. Students frequently relied on each other for feedback and help designing content. Although the assignment was nominally an individual one, by this time in the course students had built relationships with other students and tended to work together to solve technical problems and review design and content. Thus it tended to be collaborative as well as individual, as two interview comments by students show:

> I'd never worked on something on that scale. Henry helped me on the Flash. A few people in the class helped me on the nitpicky details. Matt helped out a lot. I had a lot of trouble with FTP, and just uploading it on the server, because Matt had done most of that on the genetic engineering project. So I had to start from ground zero.

> I loved that project. It was really, really time-consuming, but it was fun time-consuming. There were people you already knew who could help you on the project.

Portfolios and Self-Evaluations. At the end of the course, students submitted portfolios along with a 2-3 page self-evaluation. The object was to encourage reflection upon the student's learning trajectory and consolidate the experience of the course. Initially, I allowed students to maintain their course diaries ("Observations" notes) in a module of the Collaboration Center built for that purpose, but I dropped it due to privacy concerns, which will be detailed below.

At the same time as I was developing learner-centered modes of teaching, I was continually developing the Collaboration Center to support them. The Collaboration Center thus reflected and demonstrated my pedagogical values more thoroughly than if I was using software created by an external vendor. In the next section I will describe this software in some detail.

5. A Functional Description of the Collaboration Center

The Collaboration Center consists of 19,000 lines of ColdFusion code, which runs on a ProLinea 1600 server (400-MHz Pentium II) donated to the Computer Writing and Research Lab by Compaq. Every page the user sees is generated on-the-fly from an Access97 database.

The user interface consists of a toolbar at the top of the browser window. As the diagram shows, students see only the first two rows; instructors and administrators see more.

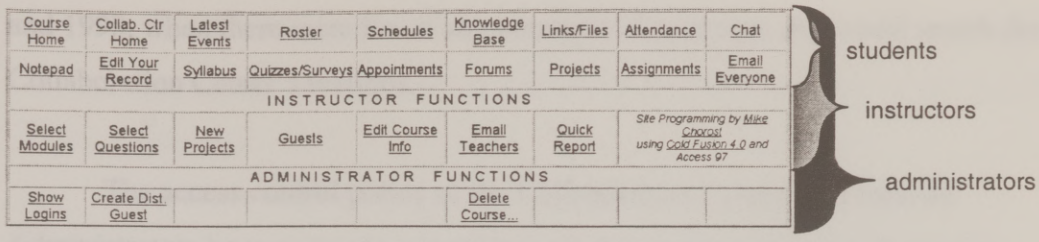


Figure 4: The toolbar of the Collaboration Center. Students see the top two rows; teachers see the top three; administrators see all four.

In terms of application design, this is a flat structure, e.g. most of the functions are available directly from the toolbar rather than a level or more down a hierarchy. While I would not claim that this is an optimal design, I found that users generally appreciated its simplicity. Students indicated to me that although the plethora of choices was initially confusing, they soon learned what was where.[3]

I made a deliberate choice not to develop a more sophisticated user interface. I chose to focus on the functionality of the modules rather than the program's look and feel. In my particular situation, I could get away with a less than optimal interface, because I was available to help students and other teachers when they had difficulties. I could amend the insufficiencies of the design by my personal support, a luxury not granted to developers of commercial software. This kept me free to focus on writing new modules and improving existing ones. Thus, while the design was not particularly slick, the program nevertheless offered substantial and robust functionality. This was an important consideration in 1997

[3] However, there is one important drawback. One has to scroll down to see most of the contents of each module, which means that the user frequently is forced to scroll back to the top to recover the toolbar. This makes for a lot of scrolling up and down. A better solution, which several students have suggested to me, would be placing the toolbar vertically in a left-side frame.

and 1998, when there were few if any commercial packages that could match the Collaboration Center's features.

The access control policy of the Collaboration Center is as follows. Administrators have access to everything: all data and functions in all courses. Since I was the only administrator, this worked well enough. Teachers see all information in their own courses. Students see all information except that which I defined as "private," such as the attendance records of other students. Finally, guests have limited access to all courses. Guests require no login; they can simply enter a course and look around. Guests cannot see students' last names, phone numbers, email addresses, and other personal information; they also have no access to data such as attendance records. They have no rights to delete anything, although they are permitted them to add messages to online forums (none ever took up the implied invitation to participate in class discourse).

I created a special category of guest, "Distinguished Guest," for guests whom teachers wish to grant a higher level of access. Distinguished Guests have essentially the same privileges as the students in the course.

Access control is enforced rigorously. Each page checks that the user has valid access to it before displaying anything; this prevents users from attempting to gain unauthorized access to particular pages by modifying parameters in the URL. User logins also time out after an hour of inactivity.

The Collaboration Center's functions include:

- An announcement space.

- Email-your-class and email-your-workgroup functions.
- An online syllabus. It can automatically inform students of changes via email.
- A roster of the students, with optional photos of each.
- Reports with summary statistics of class activity.
- A meeting-time finder which helps student groups find what free times they all have in common.
- An appointment scheduler.
- A Knowledge Base, where teachers and students can store textual information.
- Online attendance and assignment records.
- Threaded discussion forums.
- A real-time chat applet.
- File libraries where students can upload and download files (presently inactive due to security concerns).
- Link libraries where students can store useful URLs.
- "Notepads" where students can store unlimited amounts of working text (e.g., drafts and notes).
- "Project" pages for student workgroups, each with its own file/link library, mini-roster, notepad, and more.
- Survey and quiz builders that all members of the class can use.

Between Fall 1997 and Spring 1999, a span of four semesters, the Collaboration Center served 427 students and recorded over 17,000 logins through 4/11/99. In the Spring 1999 semester, the last in which I taught, it supported 10 courses on a daily basis. Most of them were at UT-Austin; two were at Lake City Community College in northern Florida, taught by Jan Tucker.

During the last two semesters of its use, Fall 1998 and Spring 1999, the Collaboration Center kept accumulated records of the actions performed by its users.

Forum messages posted	987
Student rosters edited	798
Changes of personal notepads	686
Links uploaded	461
Emails to project groups	427
Appointments made	402
Changes to group notepads	319
Teacher updates of syllabus	286
Emails to whole classes	242

Table 1: Actions Performed By Users

These attest to the fact that the Collaboration Center has performed a substantial number of functions for its users. For example, nearly a thousand Forum messages have been posted, and students have used the email-your-group function over four hundred times. The Collaboration Center also recorded 657 guest logins between 1/31/98 and 4/7/99, an average of 11 guest visits a week across all courses.

The Collaboration Center thus served many more people than in my own classes. I've focused on my own classes in this dissertation, however, because I was most familiar with them.

6. Using the Collaboration Center to Support Learning in Complex Environments

In Chapter 1, I described three key aspects of complexity theory: locality, network topology, and the utility of ecological metaphors. I explained how each of them offers models for managing work environments. Here, I will discuss how I experimented with creating material and virtual learning environments based on theories of constructivism and complexity.

In the "Locality" section, I will discuss how I used the Collaboration Center to devolve power from the teacher to student teams and individuals. I will discuss these topics:

- Consolidation of teamwork (team rooms, email functions, and instant messaging)
- Shifting the locus of discipline from teacher to student (attendance module)

In the "Network topology" section, I will discuss how I used the Collaboration Center to encourage multilateral contact and decision-making among students, and examine the consequent social issues that arose. I will discuss the following:

- Online identity (roster module)
- Communication between teams (team rooms and announcement spaces)
- Activity tracking, student privacy, and audience concerns (latest events and notepad modules)

In the "Ecology" section, I will discuss my efforts to create a richly resourced environment where students have access to high-quality information on which to base their decisions and actions. In this section, I will discuss the following topics:

- Self-organizing teams (forums module)
- Dynamic syllabi (syllabus module)
- Sharing of narratives (knowledge base module)
- Opinion economies (survey module)

It took time for both my pedagogy and software to evolve, each informing the other. My theories of how to support learning in complex environments evolved gradually over four semesters of teaching. As Donna Haraway (1991) writes,

> The boundary is permeable between tool and myth, instrument and concept, historical systems of social relations and historical anatomies of possible bodies, including objects of knowledge. Indeed, myth and tool mutually constitute each other (p. 164).

To use Haraway's terms, the "myth" of autopoetic theory and the "tool" of the Collaboration Center mutually constituted each other in my work. From the outset, the technology of database-driven Web applications made it intuitively clear that information could be decentralized and continuously updated to mirror human activity. I started with the intuitive sense that the Collaboration Center could be used to support a more organic mode of teaching, and made those intuitions explicit over four semesters of research, teaching, and software development.

Locality

Complexity theory strongly emphasizes local functional groups. In Fisher and Fisher's managerial terminology, they are called "self-directed work teams." Instead of being dependent on an outside leader for direction, they generate their own goals, for the most part, and pursue them in ways of their own choosing. The job of the teacher/manager is to set clear performance expectations, intervene strategically when needed, and provide a resource-rich environment. In this section I'll describe how I used two of the Collaboration Center's modules, team rooms and the attendance module, to devolve power to teams and individuals.

The Functions of the Team Rooms

To function effectively, teams need resource-rich workspaces. I therefore created a "Projects" (team room) module which gave each team its own online home as a space for storage and discourse. The module was inspired by the spatial metaphors which underlie the design of CVW (Collaborative Virtual Workspace). Each team room offered a roster of its members, an announcement space, an email-the-group function, a scheduler utility, a link library, and a notepad (a text storage space). During the Fall 1997 semester, the teams also had a file library, which had to be removed in subsequent semesters due to security concerns.

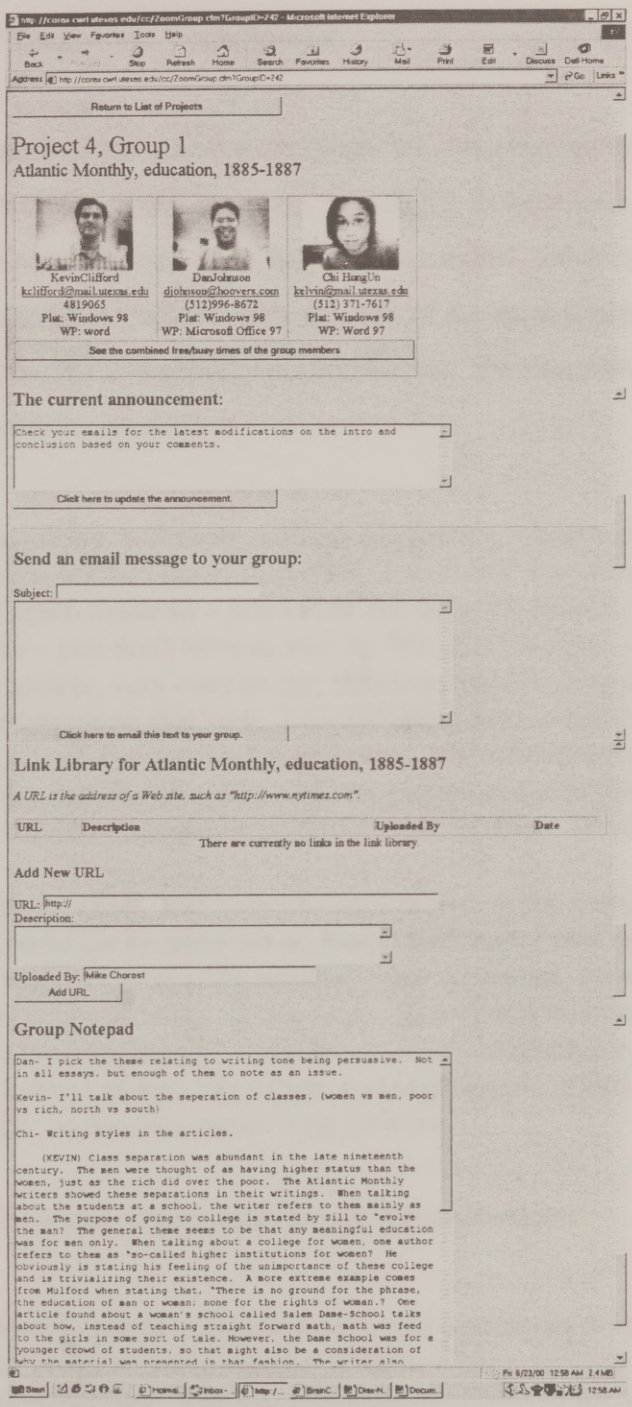


Figure 5: A project team room. Note that the team members can click on a bar to see what free times they all have in common for scheduling a meeting.

Consolidation of Teamwork: Team Rooms

A key benefit of the online team rooms was consolidation: the ability to give teams a spatial focus for their work. The ability to keep four or five people's material in one place enhanced the cohesion of the team – made it, literally, more "local" to itself and more unified. This was one of the most appreciated and remarked-upon benefits of the Collaboration Center. Two students made the following comments in interviews:

> The neatest thing, I thought, was the file library, where we could share files – I liked that MUCH better than FTP. I wish that was used in other places. It was so much easier to – you know, the default application for the Internet is the Web – you don't have to start up WS_FTP or Fetch or whatever; it was very simple, very easy to use, intuitive, and it was simple to start sharing information that way. I started using it for other things. I stopped carrying around floppy disks at that point, and started using that option to store all my data for this class. All my papers and everything, and wherever I was, I could download it and stick it back up there. It was great.
>
> After I left the class, and I worked on a project at the company I work at, I would be at a loss because we no longer had them [the Collaboration Center's tools] anymore; the notepad, your scheduler, appointments, project assignments, go and look up our team. [It created] a sense of like, darn, we no longer have these tools…you know, a central place for your group to come, reference things, file things, message board…it felt like we were going back, backwards, rather than forward.

A teacher noted how one of her students used the team room:

> Palma chose to put her favorite poetry and comments in the project area.… I think the projects area gives the students a place to keep all their information together. The links/files alone is hard to see and sort through, so having the project area allows students to access everything they need for their research.

As these comments show, the team room enabled teams to keep all their material in a single unified place, and they could access it from any computer they happened to be using at the moment. It helped make the Collaboration Center a site of "anytime, anyplace" work.

Communication Within Teams: Email, Instant Messaging, and Group Notepads

The team rooms allowed students to send email messages to the other members of the group with a single click, another frequently used function of the Center. One student related how his group used it to discuss project work in a highly collaborative way:

> A lot of times, you'd get replies from people who'd kind of build on it, or either say, "That's good, but why don't we do it *this* way," and they would think about it. Or I'd get one reply from Tim, "Why don't you do it *this* way," and I'd get a reply thirty minutes later from James saying something else, "What about like *this*?"

My students also used ICQ, an instant-messaging application which lets users see when particular other users are online and contact them immediately for impromptu exchanges of messages. It also allows users to chat and exchange files. The Collaboration Center encouraged usage of ICQ by having a Roster field where students could list their ICQ number and handle. One pair of students used it to coordinate the final stages of a paper revision:

> For the final paper, me and Lemuel, what happened was, we divvied up the paper, essentially. Everyone did their work, but towards the last part, in the last phase of editing, it was just me and Lemuel. Nobody else had that kind of access. We were using ICQ to send messages back and forth.... We were working on how the paper was going to flow, the integration. So I told him, this has to go this way, this has to segue, and he'd go, "That's a

good idea, how about *this*, what about *this* section, where does the segue begin?" And so we discussed things like that on ICQ.

Within the team rooms, students used the notepads as spaces of communication and commiseration. The following items are text from some of the team notepads:

> ****Hey you guys, I looked at our main page on my computer (in Netscape, 15" monitor) and the frame takes over half the page, with our pages on the right not being totally seen. I don't know if there is anything we can do, but I wanted to bring it to everyone's attention. :) -Marcie
>
> Well guys, I can't download YET. I am working on it..... carol
>
> Hang in there Carol :)-Marcie
>
> Mary, great link, very interesting, loved the slide show on leadership.
>
> I looked at site on GSA. Scary thought. It says how little information they do collect from you when you access their site, but is that all? But how much can they get if they wanted to. Do we have any protection from people having access to personal information about us? Very interesting article. A lot of food for thought.

Students also used the group notepads as a fallback alternative to slow email servers, as another note showed:

> Mary, I am having trouble accessing my e-mail, it is running about 24 hours behind. My address at home is [deleted] if you should want to chat this weekend, or at night anytime. Thanks Kristi PS If you haven't noticed, I finally loaded my IQ Test that we discussed on Weds.

The team notepads only worked as communicative spaces, however, if the students had established them as such beforehand. One project ran into trouble

because one student was leaving his teammates messages in the notepad, expecting them to see them and respond. They didn't know his expectations, however, and so looked at the notepad rarely if ever, leading to misunderstandings that set the project back several days.

One reason why these functions were so often used is that they provide considerable freedom within their structure. With all of them, the teacher/manager sets up the software environment but does not manage specific team interactions. That responsibility devolved almost entirely to students and teams.

From Centralized to Localized Discipline: The Effect of the Attendance Module

Locality works on the individual level as well as the team level. As Shoshana Zuboff notes, information technology has the effect of shifting the locus of responsibility downward, from the managerial level down to the individual worker. It is often the sheer visibility of the individual's electronically recorded actions, both to herself and to managers, which forces a continuing self-assessment and correction. Zuboff (1988) writes, "The psychological effects of visibility alone are enough to ensure appropriate conduct" (321). The evidence from my course suggests that this has happened with respect to attendance.

In the spring of 1997, I added a module to the Collaboration Center which displayed to each student my records of their attendance. The teacher can update the record at any time through a Java applet. A green box means "present," yellow "late," and red "absent."

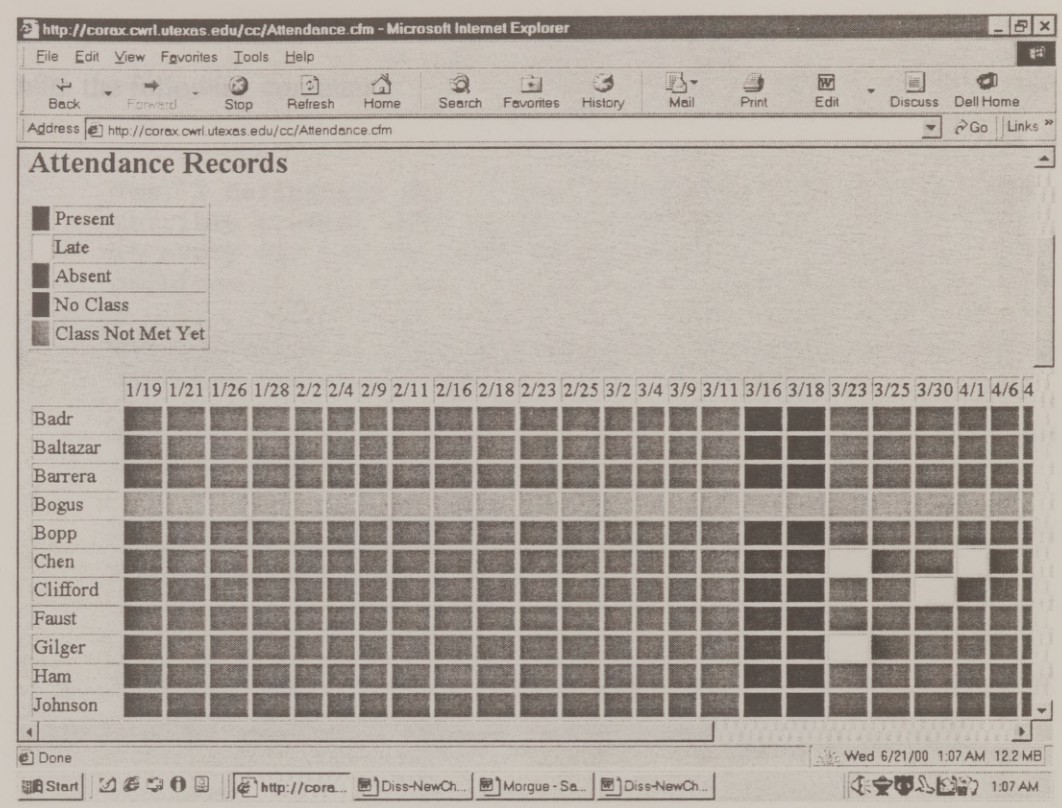


Figure 6: The Attendance module (teacher's view). Students see only their own records. The student named "Bogus" is a fictional student I created so I could view the Collaboration Center from a student perspective during development.

My end-of-semester surveys in Fall 1998 and Spring 1999 included a question about this module: "Do you think being able to see your attendance records online had any effect on your actual attendance?" Students replied with essay answers.

In the Fall 1998 class, nine out of 16 (56%) respondents replied "no," and the remaining seven (44%) replied "yes." While this latter 44% is a minority, I think it is a significantly large minority, especially given the comments attached to

the "yes" answers. Four of the students answering "yes" amplified their answer with the following comments:

> Yes, I definitely do. I mean looking at them I look like a terrible student that is why I made a vow to myself to make it every day for the last half until I ran into some problems.
>
> Well, I think it did. I knew that I didn't want that bar to look all red and yellow, so that motivated me.
>
> Yes I THINK being reminded of when I am not there I am more willing to come to class.
>
> Yes, it did. It made me think "I haven't missed any so far, why should I miss one now?"

In my Spring 1999 class, an equally large minority also replied yes – nine out of 21 (43%) respondents. Of those nine, eight amplified their answers:

> Yes - I am prone to skipping class, and this was one additional measure to keep that in check. Commitment to fellow project members was another.
>
> Yes, most certainly. It reflected my attitude toward the class and it wasn't too nice to see a flaring red or yellow colored box amidst the row of green's.
>
> Yes, because you could actually count how many days you missed, instead of setting up an appointment and bothering the teacher to figure out how many days you had missed.
>
> Yes. It looks terrible to see the colorful blocks.
>
> Yes. I didn't want to be absent because it would be a red box in a field of green. I somehow didn't miss a class; I was tempted to do a few times but I always came no matter what. I usually was cheered up by coming to class and seeing some new friends I had met here.
>
> Yes. Because sometime I forgot how many days I had missed.

> Maybe. I never missed a class because I didn't want to see any other color but green in the box.
>
> Yes. And then no. I knew that attendance was part of a grade, but I had a mental note of where I always stood in the class.

Even several of the students who replied "no" pointed out that the availability of the record increased their ability to know where they stood:

> No, but it made sure everything was being recorded correctly. Gave me peace of mind.
>
> Probably not. Although I liked being able to check my attendance just to make sure.
>
> No, but it was useful.
>
> Not personally, but it was nice to be able to look and see how my attendance was without having to keep track of it myself or get in contact with you to find out.

It is interesting to note that six of the 12 "yes" answers across both semesters reduce attendance, rhetorically speaking, to an issue of color ("I never missed a class because I didn't want to see any other color but green in the box.") Perhaps this points to the power of the collecting instinct: the desire to accumulate a consistent set of objects, relatively irrespectively of their actual value or meaning. If that is the case, then it suggests that the collecting instinct can be deliberately harnessed to achieve socially useful effects.

A similar effect appeared to happen with the Quizzes module, as used for a course that another teacher (Jan Tucker) was teaching. Until the first week of March 1999, midway through the semester, her students had no way of seeing all their test scores summed up in one screen. Then, at her request, I wrote a function to provide them with that information. Within a week after I wrote the function,

virtually every student in her course had caught up on their untaken quizzes. As with the Attendance module, I suspect that the visibility of the scores prompted students to monitor their performance more carefully.

My hypothesis is that the visibility of the information transferred the locus of discipline from the teacher to the student, e.g., moved it from centralized (teacher) power to internalized (student) power. I would suggest that the module illustrates Foucault's theory of the distribution and localization of power, as set out in his discussion of the panopticon in *Discipline and Punish* (1977).

> Hence the major effect of the Panopticon: to induce in the inmate a state of conscious and permanent visibility that assures the automatic functioning of power. So to arrange things that the surveillance is permanent in its effects, even if it is discontinuous in its action; that the perfection of power should tend to render its actual exercise unnecessary; that this architectural apparatus should be a machine for creating and sustaining a power relation independent of the person who exercises it; in short, that the inmates should be caught up in a power situation of which they are themselves the bearers (p. 201).

The Attendance module, like the panopticon, is an architecture of power. It reinforces an ethic of behavior without the teacher's direct involvement.

However, the analogy between the panopticon and the attendance module is only partial. For one thing, students are hardly prisoners (although their presence on campus is strongly conditioned by social expectations and economic imperatives). For another, the visibility of the attendance module is two-way, not one-way; the teacher's formerly inaccessible data are now available to the student, whereas in the panopticon the guard's presence is always a secret. And finally, the attendance module's power does not depend on the student's psychological

uncertainty. Just the opposite: the data are always collected and are always there. The attendance module's power is architectural, like the panopticon's, but it is not repressive. To the contrary, it is liberating to the student to know exactly where he or she stands.

Of course, the informating power of software can easily be made punishing and repressive. When I originally wrote the Attendance module, I considered making *all* the data available to *all* the students – in other words, everyone would be able to see everyone else's attendance records. I thought that the public embarrassment of having red boxes might be salutary. A talk with Peg Syverson, one of my thesis advisors, quickly disabused me of this notion. Dr. Syverson felt that attendance records should be considered private between teachers and students. I also realized that public shame would be counterproductive; it would only generate resentment and fear. As the above survey responses show, simply giving a student access to her own data was both effective and humane.

An interesting coda to this decision can be seen in one student's response to the attendance question:

> No…absolutely not. It reveal too much information. Sometimes, I could not access this page in public space.

The student seems to have misunderstood the question, as if it had been about whether it influenced him or her that *other* students could see the record (which of course they couldn't, except by looking over his or her shoulder). But the answer reveals a deeply proprietary feeling about the record. It also reinforces my thesis that the student's own perception of the record is what really counts.

All of these are ways in which my software acted to increase locality, that is, to "leverage the study of the capable, responsible, autonomous individual to act quickly and effectively" (Bair, 1997, p. 4). The consolidation afforded by the team rooms enabled groups to act autonomously. The team notepads, the email-your-group function, and ICQ allowed for collective discussion. The attendance records distributed the locus of discipline from the center (the teacher) to the individual students.

Network Topology

In Chapter 1, I drew on Castells, Kahn, Fisher & Fisher, and others to identify the key aspects of a network topology. A network consists of autonomous groups which intersect with but do not control each other, and there is an implicit focus on know-who as a crucial organizational skill. Networks need, among other things, robust communications tools to facilitate point-to-point contact and know-who for finding resources. However, the very robustness of these tools opens up a new set of problems and possibilities in the classroom, namely the issues of privacy, tracking, and audience.

Creating an Online Identity: The Dynamic Roster

The roster module was developed to enable students to present an "identity" to the class and update it as that identity changed. It allowed students to see not only each other's basic biographical information such as photo, email, and phone number, but also other data such as type of computer, word processor, and an "interesting fact."

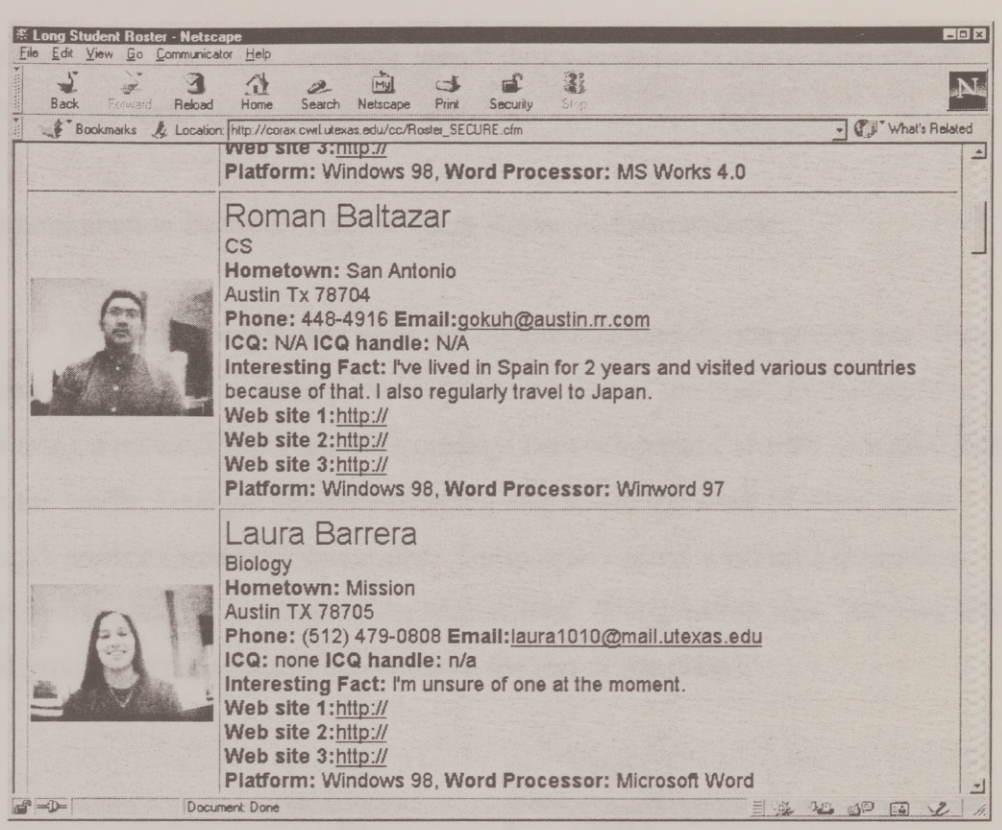


Figure 7: The roster (student view). Teachers who view the roster see all the student's information. However, students have full access to their own information.

On the surface, the roster looks no different from a static Web page. (Before I learned ColdFusion, I maintained a very similar roster as a hand-coded HTML page). However, a database-driven roster is more suited to a network topology than a static page. It can be updated at will by students themselves, without having to pass through a central authority. Many students used it as a space of self-presentation, by entering a witty or provocative fact about themselves in the "Interesting Fact" field. It can be said, then, that the Roster is a collectively produced document, a space of self-presentation and exploration. This makes it

well-suited to a network topology, which depends on the ready availability of information about the actors in the network.

Communication Between Teams: Team Room Announcements

While the team rooms were principally intended for the use of their own teams, I also chose to make them visible to the rest of the class, in the hopes of fostering a network topology via exchange between groups as well as within them.[4] On the whole, students appreciated being able to see the work of other teams, though some expressed reservations. These reservations were articulated in a survey I conducted on usage of the team rooms. The question was, "Do you feel that group notepads should be visible to the rest of the class?"

Yes	11	(55%)
No	5	(25%)
No opinion	4	(20%)

Several students amplified their answers with the following comments:

> depends on the assignment. the survey would be bad to see ahead of time.
>
> It's hard to say. I'd like to be able to see what other people are doing, to rate my work, but in the same breath, I don't want other people reading MY work...
>
> You may get cookie cutter work.
>
> It may lead to a lack of originality.

---

[4] In fact the code initially allowed any student to make changes in any project room. Several of my students pointed out that this could let someone accidentally delete another group's material, so I revised the code to prevent that.

> Even though you don't want it to happen, some groups ideas and thoughts will manifest themselves in your writing.
>
> Would be easie to get the jist of what others were doing.
>
> Although input from other groups is sometimes useful, I think this was part of the reason we saw so many "similar" traits in our ideal environments.

As the comments show, some students wondered if the public nature of team rooms would lead to "cookie-cutter" work where teams imitated each other's production. One student, writing perhaps tongue-in-cheek, noted the downsides as well as upsides of an open environment: "I'd like to be able to see what other people are doing, to rate my work, but in the same breath, I don't want other people reading MY work..."

Despite these reservations, slightly more than half of the students (55%) wished to keep the team rooms visible, and 20% offered no preference one way or the other. Personally, I felt it was important that students be able to observe what was going on in other teams as a way of comparison and normalization; it helped keep groups from becoming too isolated.

The team rooms' announcement spaces were ostensibly for use only within teams, but since I made them public to the course, they became spaces of humorous presentation and banter shared with the rest of the class – another example of network topology in action. Most of the banter was verbal, as seen in the screenshot below, but some of it was pictorial, as seen in the screenshot after it.

Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Stop

Bookmarks Location: http://corax.cwrl.utexas.edu/cc/DisplayGroups.cfm What's Related

Group 1	Technology in Schools	Blaine Bybee Heather Weathersby	Meetings are at 6 on Tuesday and 7 on Wed.
Group 2	Distance Education	Lemuel Haham Karen Meyers Samuel Quek	Final draft due at 9pm to me (Samuel) on Wednesday. I am so happy we finished this paper. It was exhausting.--Karen
Group 3	Journalism on the Internet	Jana Colletti Mark Dulyunan Farah Homsi	Rough Draft is posted in file library. Post a message here on the web if you have any problems and i will put it on the notepad from California. I was trying to put it on the file upload to keep our notepad from being clutered. -Jana
Group 4	Microsoft	Mark Lassoff Kristen Lea Evan Ochs	4:42 AM.. Done Deal.. No problem :)
Group 5	Cloning	Amber DeSutter Tar Patel Wai Fong Tsang	Good job everybody! We're finally done! Amber
Group 6	Webgrrls	Christi Conway John Garcia Dean Yates	The final is done :) !!! Webgrrls don't rule anymore.

Document: Done

Figure 8: Banter in the team room announcements.

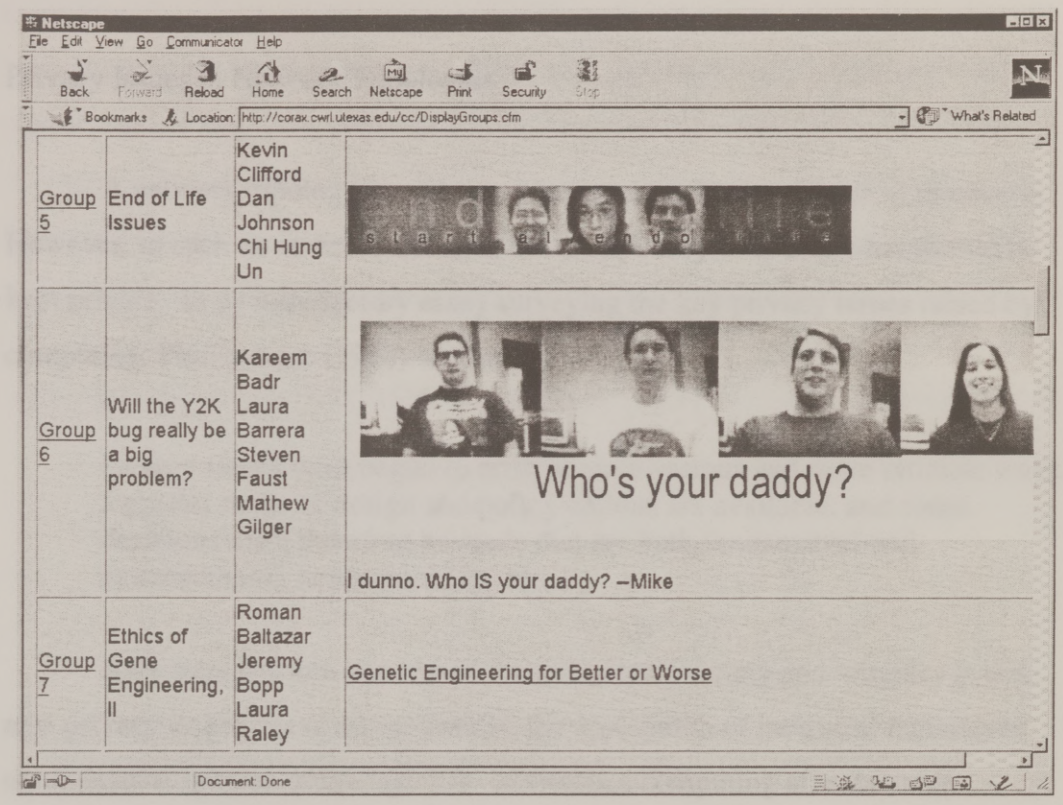


Figure 9: More banter in the team room announcements.

In the second screenshot, it can be seen that students copied pictures out of the roster module and incorporated them into graphic elements, which they posted on their own Web accounts. They then inserted hyperlinks into the announcement space to make the graphics appear. Some of the graphics rose to the level of commercial art, e.g. the graphic for the "End of Life" team.

As the screenshots demonstrate, the banter ranged from casual humor to sophisticated graphics. But whether simple or complex, it served the function of letting the team construct a particular way of talking within themselves and showing themselves off to the other teams.

Privacy Issues in Network Topologies

A network topology implies openness and the free exchange of discourse. However, in such an environment, it is not always easy to decide what should be kept private. In an introductory essay surveying the key privacy issues raised by computing, Phillip Agre (1998) writes:

> Privacy issues have begun to arise in more various and more intimate ways, a greater range of design and policy options are available, and some decisions must therefore be made that are both fundamental and extraordinarily complicated (p. 6).

Agre suggests that one reason for this greater variety and intimacy is that new privacy issues "have arisen through the application of industrial methods to nonindustrial spheres of life" (p. 57). Networked computing affords richly multilateral topologies of interaction, but it also creates challenging ambiguities about where the boundaries between private and public should lie. In the next two sections I'll examine activity tracking, which makes individual actions visible to other participants in the classroom environment, and the notepad, which during its initial testing raised issues of privacy and audience to such an extent that I ended up not using it for its original intended purpose of storing student diary entries.

Activity Tracking: The Latest Events Module

In his article "Beyond the Mirror World," Phillip Agre (1998) examines a particular kind of representation of human activity – the development of notational systems which track, and later reveal, sequential individual actions. The concept of the mirror world, originally invented by David Gerlenter (Bennahum, 1996), refers

to a computing domain which faithfully represents actions performed by individuals in the course of their daily lives. The latest events module is a "mirror world" kind of application, because it displays a list of the last 48 hours' worth of actions performed by the class. The point of the module is to reveal activity as it happens, to create a sense of the "ongoingness" of the site. It also alerts users to changes without their having to examine the modules themselves to see what's new.

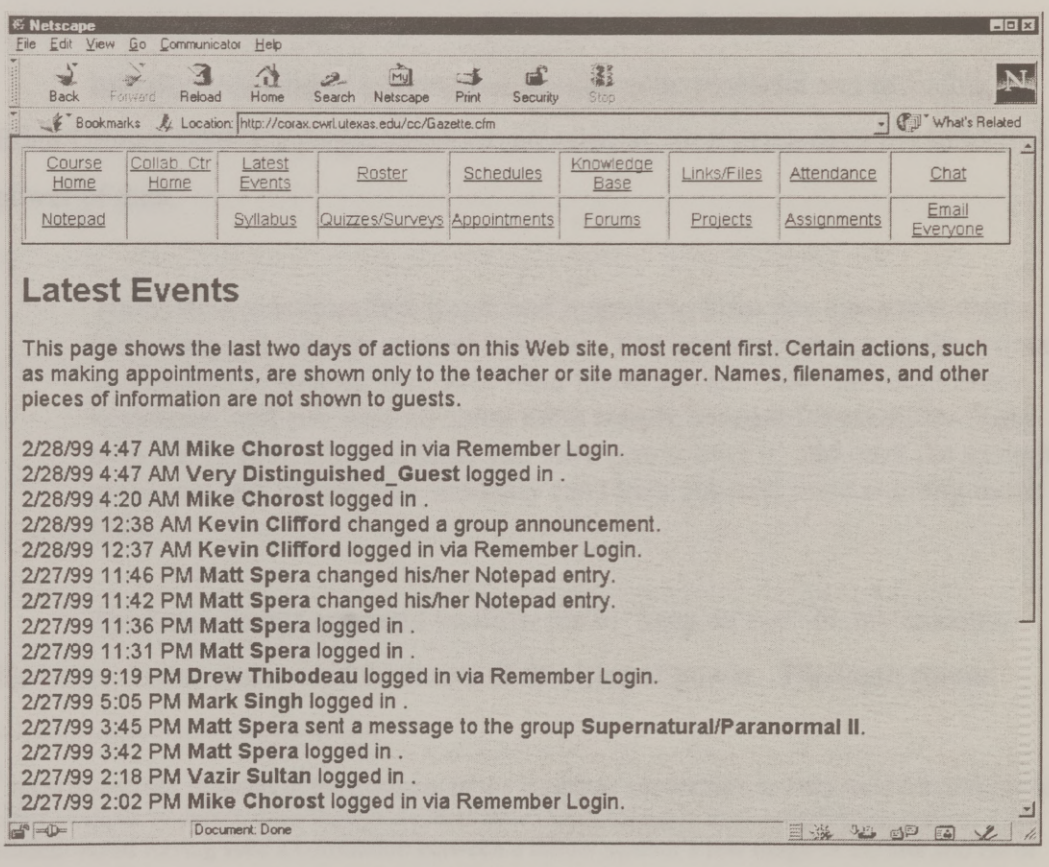


Figure 10: The latest events module. Events are listed in descending order, most recent first.

The functionality of this module changed significantly over time. Originally, it simply recorded the number of times a student logged into the

Collaboration Center. In this section I will tell the story of how students reacted to those counts and relate how I modified the module to address their concerns.

As originally written, the module showed teachers each student's name and the number of times they had logged into the Collaboration Center. When students viewed the page, they saw only the numbers, in a scrambled order so that the counts of individual students could not be deduced from it.[5]

Initially, this kind of information seemed to be powerful and revealing. Ellen Ullman (1997), a programmer turned essayist, nicely describes the seductive power of data.

> The system was installed, it ran, and it spoke to him: you can know every little thing you always wanted to know. You can keep an eye on the woman you trusted to pick up your kids from kindergarten. You can count every keystroke, and you want to count them simply because it's possible. You own the system, it's your data, you have power over it; and once the system gives you this power, you suddenly can't help yourself from wanting more (88).

The login counts appeared to allow me to "keep an eye" on my students. However, I soon discovered the limits of this kind of power. The login counts

---

[5] As it turned out, students at the extremes of the login bell curve (high or low) were not difficult for other students to detect, even though students didn't have access to each other's login counts. One student noted having seen a correlation between a fellow student's low usage of the site and his or her low performance:

> ...I do remember there were some people who never logged on. I think we all knew who those people were. I think we all pretty much knew the extremes...so in that sense it was frustrating to know that one of my classmates would not make use of the site. I can see how that would affect his performance so much and there was so much going on on the site that you would be totally out of the loop if you weren't there. This guy had visited, like, once, by midsemester, and it totally blew me away. It made me more critical of my peers.

revealed quantity but not quality. One student might log in once and keep her login going throughout the day, whereas another might be moving around campus and log in four times in the same period to do the same work. I soon discovered that the login counts were only meaningful at the extreme ends of the scale: some (but not all) low-performing students logged in relatively infrequently, and some (but not all) high-performing students logged in very frequently. But even in those cases, the numbers told me nothing about the students that I did not already know from their written work and class participation.

For their part, my students had their own reactions to the fact of login tracking. In my Fall 1997 class, the first which had login figures available to them, there was a wide diversity of reactions to the counts. (At this point the Collaboration Center consisted of only the Roster, the Forums, the announcement space, and the login counts). My class undertook an online chat session on the topic of tracking of online activity. The discussion was contextualized by a reading from Neal Stephenson's cyberpunk novel *Snow Crash*, which presents a parodic but all-too-plausible scenario of employee surveillance. Student attitudes toward the counts varied dramatically, from indifference to indignation. Here are several representative contributions to the dialogue.

> I stopped using the website because I want the anonimity of not being tracked. I'd rather any ideas ands thought I have not be succeptable to mikes magic marker. I work around it buy not giving thoughts and ideas over the web.

> The first thing that I thought when Mike mentioned how everything we did in the class website was recorded was "Gee, maybe I should log in more so I can raise my statistic." But soon after I thought that I realized that numbers like that don't say anything about how much work we actually do - I never turn my computer off and sometime one loggin for me can stretch on for days. So I didn't modify my behavior after all.

> I have to admit that 'numbers' fascinate me. When I learned that I was being watched as to how many times I logged into the collaboration center, I would log in unnecessarily, even if I was just on the page to check the course schedule, something which did not require logging in.
>
> I think being watched definitely affects how I act. I didn't really have any objections to Mike surveying us because the information was not personal and supposedly is not supposed to affect our grade. But i can see people trying to manipulate data to make them look good.
>
> Yes, it changed the way I behaved. I would go to the class web site to see what the plan was for class and stay caught up. However, the surveillance made me want to be more actively involved. I do not know most of the aswers asked on e-mail so it was difficult for me to answer anyone, but I started e-mailing the listserver, Mike, and people in my group instead of asking in class or calling like I had been.
>
> i didn't see that many of those numbers had much validity at all, so there seems to be no point in worrying about them. i just do what i have to do without worrying about who's watching.

I responded to my students' reservations in two ways. First, I told them about my own discovery about the login counts, that they told me very little that I didn't already know. Second, more fundamentally, I decided to change the way the tracking was done. I wrote new code that shifted from a focus on *logins* to a focus on *actions*. This proved to be a far more meaningful form of tracking. Instead of a list of names and numbers, it was a narrative, a story of actors and actions, as can be seen in the screenshot of the latest events module. Instead of seeing aggregate numbers, users see information about what has happened recently. This kind of information seems less intrusive and more meaningful. It is information embedded in a social context.

As for the code that did the login counts, I retained it, but moved it to a module visible only to teachers. Personally, I rarely looked at it after that; it is not known how often other teachers looked at the information or how they interpreted it.

Did the new approach to activity tracking address my students' concerns? I believe so. My end-of-semester surveys show that between the Fall 1997 and Spring 1999 semesters, a span of four semesters, increasingly larger numbers of students replied "no" to the question of whether the tracking tools affected when and how they used the site.

The question was phrased as an essay question in the end-of-semester surveys I conducted each semester. In Fall 1997 and Spring 1998 the question was phrased this way:

> Did your knowledge of the monitoring tools (i.e. the recording of your logins and activity) change how often you used the site, and/or what you did on it?

In Fall 1998 and Spring 1999 I reworded the question slightly, to reflect the new Collaboration Center's ability to record specific actions:

> As you know, the site records your logins and many of your actions. Did this affect how often you used the site, and/or your actions?

Over four semesters, 73 students took the survey. I have analyzed the answers by coding them on a four-point scale, viz. "Strong Yes," "Weak Yes," "Weak No," and "Strong No." Here are examples of answers that I coded on this scale.

Strong No: *No, it really had no effect.*

Weak No: *Not really but I did feel like I was being monitored and it made me feel a little apprehensive about the times I hadn't logged on in a while.*

Weak Yes: *A little maybe. Sometimes I just needed to get some info so I would log in quickly to get both accomplished.*

Strong Yes: *After I found out about the recording of the logins etc., I felt compelled to log in more and click on everything in the site because I didn't want to fall behind compared to the class.*

Answers like "not really" and "a little maybe" were usually coded as "weak no" and "weak yes" answers. (The complete set of answers is given in Appendix 1.)

Once the answers had been coded in this fashion, a clear trend became evident. The number of students reporting that the tracking tools affected their behavior dropped dramatically between Fall 1997 and Spring 1999.

	"Did your knowledge of the monitoring tools (i.e. the recording of your logins and activity) change how often you used the site, and/or what you did on it?"					
	Strong Yes	Weak Yes	Weak No	Strong No	Other	Total
Fall 1997	2 (11.8%)	5 (29.4%)	2 (11.8%)	7 (41.2%)	1 (5.88%)	17
Spring 1998	2 (11.1%)	3 (16.7%)	4 (22.2%)	7 (38.9%)	2 (11.1%)	18
	"As you know, the site records your logins and many of your actions. Did this affect how often you used the site, and/or your actions?"					
	Strong Yes	Weak Yes	Weak No	Strong No	Other	Total
Fall 1998	1 (6.25%)	0 (0%)	1 (6.25%)	14 (87.5%)	0 (0%)	16
Spring 1999	1 (4.55%)	1 (4.55%)	3 (13.6%)	16 (72.7%)	1 (4.55%)	22

Table 2: Responses to survey question on activity tracking versus behavior.

As the numbers show, the percentage of students reporting a "Strong No" – that tracking did not affect their actions – increased from the 30-40% range to the 70-80% range over four semesters.

Why this dramatic shift over time? I have a number of hypotheses.

More Skillful Framing. I framed the issue of tracking more skillfully as the semesters went on, by explaining in detail its value and use. During an interview with me, one student in my Fall 1997 class, the first in which I implemented user tracking, commented that my presentation of the login details frightened a number of students:

> *Student:* Here's the way it happened at that time…we only had maybe three people who knew a lot about computers in that class. There's a lot of amateurs who didn't know what it was…There were only two CS people. You introduced the Collaboration Center, which you know, I was fine with, but a lot of people in the class were like, "Uh, yeah, I don't know what's going on." And like, two weeks later, in class…you just casually said *[spoken aside to the other student participating in the interview]* he doesn't even realize it's going to be an issue, he just mentions he's been tracking…
>
> *Interviewer:* Oh, I knew it would be an issue.
>
> *Student:* Well, you didn't seem like you knew, you just casually mentioned it. Everybody in the class was like, "oh, my gosh!" and then, when you get groups of people, people feed on each other. One person was like, "You were tracking us!" and like somebody else was like, "Oh, I can't believe you're doing that!" So you said, 'Okay, I'll put it on the web site, but I won't put anybody's names on it.'

This first presentation of login data happened relatively late in the semester, and it did indeed come as a surprise to the class. However, in subsequent semesters I gave the students a tour of the Collaboration Center in the first week and carefully explained what information was accumulated and why. In particular, I explained that while the tracking revealed the *quantity* of usage patterns, it revealed nothing about the *quality* of students' work while logged in. I also told them that individual login counts could only influence their grade if they were strikingly low – if a student only logged in five times during the semester, I told them, that would tell me something. But even in that case, I would add, low numbers like that would almost certainly be matched by poor written work and class participation, which would tell me much more. Their login and activity figures provided me with only one narrow channel of information about their performance, compared to the much richer information given to me by their assignments and class activity.

I also took pains to show my students that they had access to all data about themselves. I believe this served to naturalize the technology, so that it was enfolded into the class from the start. It also gave the students fair warning in a situation where they were able to drop the class without real inconvenience. Thus there was nothing coercive about the tracking. One student noted during an interview,

> Here, I think a lot of people just came to accept that well, this is an ongoing class, here is where I do my work. If this were unrelated to class or work, I wouldn't stay on. I wouldn't want someone tracking my movement or actions. Those are *my* actions, and if I want to share them with people, *I'll* share them. But here, the whole point of the Collaboration Center was to get people to collaborate, and to collaborate, you have to know what everyone else is doing.

Students generally seemed to accept my framing of the tracking information. In the end-of-semester surveys, several students noted that they were concerned about tracking at first, but then became less concerned as time went on (these were all coded as "weak yes" responses):

> No, I forgot they existed as the class progressed. I was apprehensive about them at first, but in time I didn't notice. They didn't change how or when or why I used the website.

> Yes, at first it did because I thought you were taking it into consideration when grading, but then I found that you didn't so I didn't think about it anymore.

> At first, I was checking up constantly just to make sure that this wasn't used against my grade - a way to try to make sure I got recognition for my interest in the course. Then, as the class progressed, I'd check in only as I needed to.

> Yes, at first it did because I thought you were taking it into consideration when grading, but then I found that you didn't so I didn't think about it anymore.

I believe these comments point to a successful framing of the tracking information as a trivial factor in student evaluation, with a consequent diminishing of student anxiety about the issue.

Indifference. Students may reflect an increasingly prevailing attitude of being blasé about tracking. A student in my Spring 1999 class commented during an interview,

> For me, it's implied that, I accept, when I'm on the Internet, it's like I'm walking out on the street, and whatever I do, may be visible to anyone. I can see over time, more and more people will probably come to accept that this is just going to be the norm.

This student also noted that that the "invisibility" of tracking contributes to a kind of willful ignorance, where users may be vaguely aware that their actions are recorded but choose not to be concerned about it:

> It's going to depend a lot on the person. If it's below the horizon, no one's really going to think about it, if it's not brought in their face all the time – we're doing this with the information, you can't do anything about it – they don't really care.

This may indicate a learned helplessness – "I can't do anything about it, so why worry about it?" Alternatively, it may indicate that these matters are not considered threatening to reveal. The fact of one's logging into a course Web site, and then, say, posting a Forum message and then logging off, may not be considered particularly central to one's identity or security and therefore not problematic to reveal.

Several survey replies seemed to support this attitude:

> No, I didn't really care that my actions were logged.
>
> No. I don't care who looks at me on the Web.

One possible reason students don't care, one student suggested in an interview, is that privacy comes relatively low on the list of worries:

> [People have] much bigger things to worry about. Like getting my project done on time, that kind of thing. I'm willing to sacrifice that little bit of privacy in order to get my work done in a convenient way.

The survey suggests that this attitude was more prevalent among my students in 1999 then it was in 1997. Perhaps this is because of the greater ubiquity

of network technologies in daily life. The average student is more likely to be aware that web sites such as Amazon.com or CDnow.com record their logins and purchases.

Should we be disturbed, as teachers, by this apparent indifference, if that is indeed what it is? The question is beyond my scope here. I would simply note that a teacher could engage this issue in class and encourage students to think more deeply about it.

Trust. I believe that the design of the Collaboration Center's tracking functions, and the way I framed them in class, created an atmosphere of trust that the data would not be misused. In an interview, one student explicitly noted the importance of trust:

> If you're doing a good job you shouldn't have to worry about these things. I think as long as you're comfortable with who you're working with and who's analyzing the data, then it shouldn't be a problem if you trust that individual.

Consider, also, one of the student replies on the end-of-semester survey:

```
Yes, at first it did because I thought you were taking it
into consideration when grading, but then I found that you
didn't so I didn't think about it anymore.
```

These students apparently accepted my point that I wasn't using the data to influence grading decisions. Achieving this credibility takes more than a simple declaration, however. The Collaboration Center was designed to emphasize openness, on the teacher's part as well as the students, and my hypothesis is that this helped generate trust. When I was designing the Latest Events module, I consciously decided that teacher data would be just as visible as student data. My

ethical principle was that I should not ask students to reveal what I myself would be unwilling to reveal. The roster entry of the teacher, for example, shows the teacher's home telephone number and address to students. In fact, when I was designing the database, I made no distinction between teachers and students – everyone's personal records (name, address, email, and so on) are contained in a table named "Students." The implicit message I wanted to convey to the class was, "We're all students here," and I built that value directly into the architecture of the Collaboration Center. Furthermore, the roster is alphabetized by last name so that the teacher's name appears alphabetically in the list, instead of at the top (or not at all, which would emphasize the teacher's remoteness from the class).

I made a number of other design decisions in favor of openness. I had briefly considered writing code to suppress the fact that I would log into the Collaboration Center on Saturday nights. I rejected this possibility on the grounds that my students had the right to see my logins just as I had the right to see theirs. This was grounded in Michael Benedikt's Principle of Personal Visibility (1991), which he articulates as follows:

> (1) Individual users in cyberspace should be visible, in some non-trivial form, and at all times, to all other users in the vicinity, and (2) individual users may choose for their own reasons whether or not, and to what extent, to see and display any or all of the other users in the vicinity (p. 177).

Benedikt explicates the first part of the principle with reference to an ethical belief:

> The principle of personal visibility instills the belief that democracy, even in cyberspace, depends upon accountability, and that accountability depends in turn on *count*ability, that is, on the obligation to "stand up and be

> counted," to be there, in some deep sense, for others. An open society requires the open presence – each to the other – of its people (p. 178).

I kept this ethical imperative in mind while designing the Collaboration Center. I felt that for an online community to work, it should be open and egalitarian – that teachers should have special privileges only when absolutely necessary, and that information about presence and activity should be public except where there is a compelling reason to keep it private.

That is why I chose to resist the temptation to conceal my Saturday night logins. Upon finding out about that decision, one of my students commented during an interview:

> Just by your thinking you could write code to suppress your Saturday night logins, just by having the thought, that's how easily data collection can be corrupted, right there. I mean, you're an educator, you're a great guy, and everything's straightforward, and yet you're like, "oh, how easy it would be to suppress that." Right there, anybody, even Mike, might have done it. How many people would have suppressed it out of existence?

The fact that I could have hidden something about myself but chose not to builds trust. It is on these relationships of trust that a network flourishes. I think my students trusted me not to abuse the tracking data, and that trust enabled the majority of them to accept that tracking data would not be used to characterize their performance in a narrowly quantitative way.

However, the concerns of the minority remain significant and deserve attention. Out of the 73 students polled, six stated unambiguously that the tracking did affect their actions:

> Yes I tried to log in more times in order to appear to be visiting more than I was.
>
> After I found out about the recording of the logins etc., I felt compelled to log in more and click on everything in the site because I didn't want to fall behind compared to the class.
>
> Yes, I was more conscious of logging in. I felt like I had to or Mike would know I was not using the technology.
>
> Yes, I logged in a couple of times just to up my stats.
>
> Yes I discontinued use of the website after I found out about it. I think you should announce that you would be anonymously tracking.
>
> Yes. When I discovered how many times I logged in at the Collaboration Center home (pretty high), I began to keep the window open to the Collaboration Center on my computer for most of the time so that I could keep the number low. For some reason I felt that logging in too many times would higher expectations from the teacher. I don't know, truthfully... but it did lessen the times I refreshed or logged in.

For these students, it is clear that my assurances were either not understood or not accepted. The first four responses indicate that the students logged in more often to inflate their numbers, while the last two indicate that the students logged in *less* often. The last response is especially interesting, as it suggested that the student believed that higher numbers would increase my expectations.

How to address the concerns of these students? Rather than try to persuade them to think differently, it would make more sense, pedagogically speaking, to set up situations where students themselves could assess how tracking data gets used. For example, team rooms might keep records on when team members entered the room and what they did there, visible only to the team. After a project, students would have the opportunity to reflect on how the numbers altered their perceptions of each other. My prediction is that students would discover that their teammates'

production and interaction styles have far more impact than any numeric data. Whatever the outcome, students would be able to engage with the issues and form well-substantiated opinions about them.

Activity tracking is a complex issue because it has as much to do with perceptions, group dynamics, and framing as with technology. It will not become easier or go away, because it is part and parcel of network topologies – the ability to see and be seen. However, in this section I have shown that it can be implemented in such a way that most students do not perceive it as a threat. The key is to build openness and multilateral access into the software and to carefully frame why and how the numbers are collected and interpreted.

Who's the Real Audience Here?: The Notepad

Even though it is one of the Collaboration Center's simplest modules, the notepad raised surprisingly deep issues of visibility, privacy, and audience. The application is a simple "text bucket" where a user can store any amount of text. Each student has his or her own personal notepad, which teachers (but not other students) can see and modify.

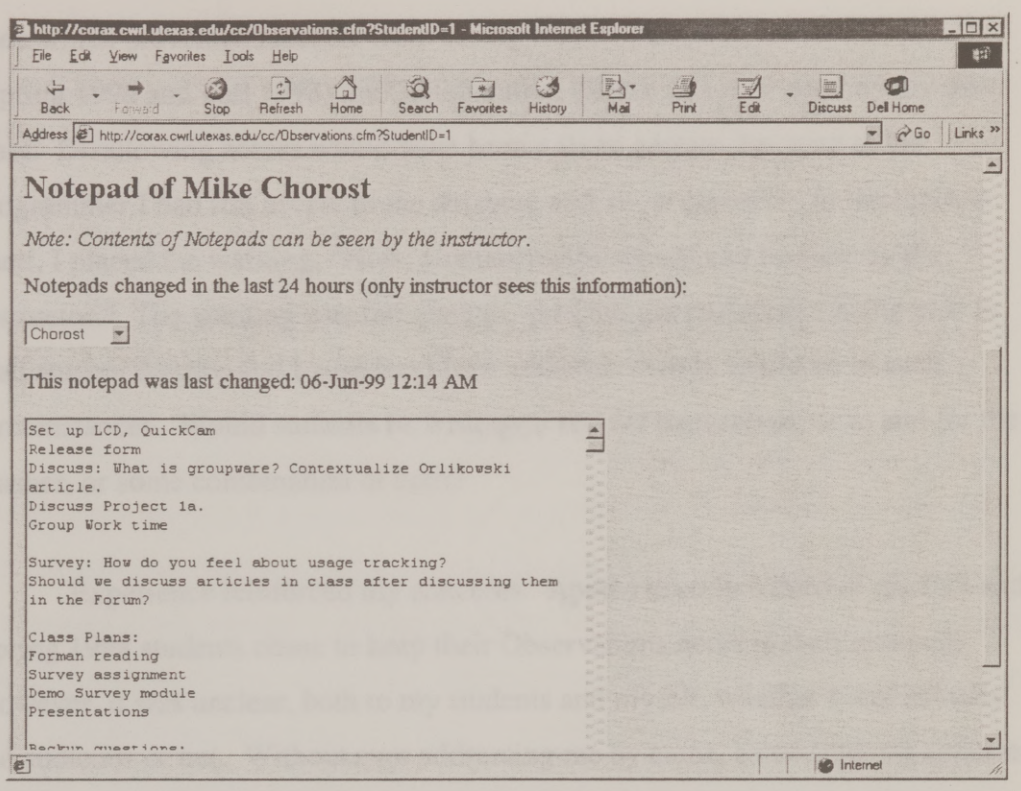


Figure 11: An example of a personal notepad.

The original idea for the module came from Learning Record theory's emphasis on personal narratives of learning. Learning Record theory invites students to keep diaries of their experiences throughout the semester, which they can then draw upon in writing their self-evaluations. It seemed like a good idea to write a module which would allow students to store their diaries in the Collaboration Center, if they so wished. It would allow them to further consolidate their work in the online space it offered.

Hence I wrote the module in the spring of 1998 and titled it "Observations." However, almost immediately, I was troubled by the question of whether the teacher should get to see the students' diaries. It would have been easy to write

code which prevented teachers from seeing student notepads, but at the time (Spring 1998 and Fall 1998) the Collaboration Center was only used by my own class. Disallowing access would have been a sham gesture, because as the programmer I had full access to the database and all its contents. In the module itself, I placed the warning, "Note: Contents of Notepads can be seen by the instructor." The warning was fair enough, yet I felt uncomfortable being able to read student diaries, and I wondered how authentic diaries would be in such circumstances. Would students be writing to and for themselves, or to and for the teacher, or some combination of both?

Experience reinforced my concerns. Approximately a third of my Fall and Spring 1998 students chose to keep their Observations notes in their notepads. However, it was unclear, both to my students and myself, whether I was actually the audience or not. Without ever addressing me by name, some students appeared to be using their notepad as a back-channel method of communicating with me:

> I don't want to sound like I'm searching for sympathy, but this is the week of midterms. I had that web page due Tues., a presentation and book due for one of my advertising classes Wed. (which I've spent 25 hours working on in the last 4 days).

It was also used as a way of raising concerns about the assignments. One student wrote,

> Well, we have turned in the paper and now have had a little break for Easter. We are about to start our web site project but Mike hasn't decided if we will work in groups or not. I hope we don't because I think it is easier when I work by myself, especially with something like this.

It was hard for me to avoid reading this message as something intended to be "overheard" by my ears. And another student used her notepad to record her worry that her self-assessment would be visible to the Internet at large:

> Not to mention the fact that my self-assessment and Picture will be on the World Wide Web for everyone to see. What if someone just so happens to stumble upon it? I don't think I want some stranger to know all the things I write in my self-assessment.

This text struck me as doubly ironic given that the student was worrying about online privacy in a space whose privacy itself was ambiguous. In this case her concern about privacy was legitimate but somewhat misdirected given that the assignment made turning in the self-evalution on paper the default option. Students could, if they wished to acquire the additional experience, post it online as an Web page with no links into or out of it to keep it hidden from Web surfers and search engines. (If I knew then what I know now about the ability of search engines and hackers to find even unlinked pages, I would not have allowed students to post their self-evaluations on the Web at all.) In any case, the concern was not addressed to me personally, and it disturbed me that the student was articulating it here instead of simply bringing it to me directly.

I was also disturbed to find that some students would use the diary entries as a way of criticizing other students:

> I was VERY upset that [Joe] didn't bother to show up or call! I spent a long time editing his part so that it would flow with the rest of our paper. [Fred] and I can not believe [Joe] still hasn't bothered to call to find out what's going on! In fact we've met five times and three of those times [Joe] hasn't showed up.

The student's frustration was fair and understandable, but I was disturbed that it was being expressed "in my earshot" yet ostensibly not to me directly. This was problematic because I wanted to be either openly involved with my students' teamwork or not involved, not at some ambiguous place in between. In a networked topology of interaction, where channels of communication are potentially numerous, it is necessary to clearly define the purpose of any given channel. Ambiguity leads to confusion and potentially destructive "back-channel talk." For that reason, I renamed the module from "Observations" to being simply the "Notepad" and recast it as a simple text storage space, to be used for anything the student wished. I occasionally asked students to post draft paragraphs there, but apart from that, it had no further role in my course.

Maintaining an Ecology of Knowledge

In the previous two sections, I discussed how I designed and used online tools to facilitate interaction among students, teachers, and teams. Since many new kinds of interactions became possible, and were actively used, the result was a significant expansion in my course's ecology of interactions and resources. The total volume of message traffic, and other out-of-class interactions such as posting and reading information, and observing the actions of other students and teams, was undoubtedly higher than it would have been without such tools. When students can log into a course Web site at 10 p.m. on a Thursday evening, note the simultaneous presence of other members in the class, and strike up impromptu online conversations with them, a new and larger kind of classroom ecology has come into existence, supplementing and enriching the paper-based and face-to-face interactions which have long been the primary medium of education. This expansion is significant because, as I argue in Chapter 1, it makes the classroom a

more authentic image of the kind of real-world environments students will encounter as they start their careers. Learning and action can now coexist in virtual and physical spaces simultaneously.

Given the complexity and multilaterality of the interactions which take place in human workplaces, ecological metaphors work better than mechanical ones for developing and describing managerial tools. The mechanical model favors images of command-and-control and hierarchy, and it casts the manager as a controller. The ecological model, on the other hand, sees an organization as a quasi-living entity, which can be perturbed and shaped but not controlled. This means that the manager's primary work in an ecology of knowledge is to focus on the overall conditions of the environment, by doing what Kimball and Mareen Fisher call "boundary management." This imperative applies as much to the constructivist classroom as it does to workplace environments.

Boundary management largely consists of managing ecosystems. In this section, I focus on tools which help teachers clarify, delimit, and shape classroom ecosystems. These tools are especially necessary, because without tools for high-level oversight and strategic intervention, the centrifugal forces of locality can dominate the class, pulling teams away from both the teacher and each other. Tools for boundary management help teachers maintain the cohesion of whole classes, so that they do not end up simply presiding over groups of groups.

The tools consist of four online techniques I developed for aiding boundary management in information ecologies: self-organizing teams, dynamic syllabi, shared narratives of experience, and opinion economies.

Self-Organizing Teams

Fisher and Fisher stress the self-directedness of work teams in a network topology. This self-directedness can extend to the forming of the teams themselves, around natural alliances of interests and likings rather than by the centralized command of the teacher/manager. This is especially important in a course where the majority of the work is project-based; as I noted earlier in this chapter, students collaboratively wrote research papers and created Web sites in teams. Team formation, then, should not be handled haphazardly or randomly. In this section I describe a process of team formation which was decentralized, yet enabled a coherent order to emerge from the individual decisions of each member of the class. The process amounts to a genuinely self-organizing system.

Before hitting on this method, I used a fairly centralized, blackboard-based method of organizing teams. I would ask students to come in with one or two ideas for topics. We would write all the topics on the board, and students would "call" their interest in a given topic or another. After eliminating topics that drew no interest, we would repeat the process with the remaining topics, until the class had "shaken down" to three to five students in each topic. I would devote full class sessions to this business, but even so the results were rarely fully satisfactory: many worthwhile topic ideas received short shrift due to time pressure, and several students always wound up in undesired groups in order to make the team sizes come out even.

In the Fall 1998 semester, two of my students, Gregory Bodle and Tim Smith, proposed that we use the Forums instead to form groups. We worked out a process where students proposed topics in Forum messages, and other students "joined up"

by adding and deleting messages, according to carefully specified rules. The rules ran as follows:

- Groups must have not less than three nor more than four members to "make."

- You may join more than one topic, but once you finally decide which topic you really want, you must remove your joining-message from the other topics.

- If you stay with your own topic but too few people join you, then you'll need to leave your topic and join someone else's.

- If you are in a topic with too few members by [date], you must move to another topic.

- You may, if you wish, leave your own topic at any time. You are not obligated to remain in your own topic.

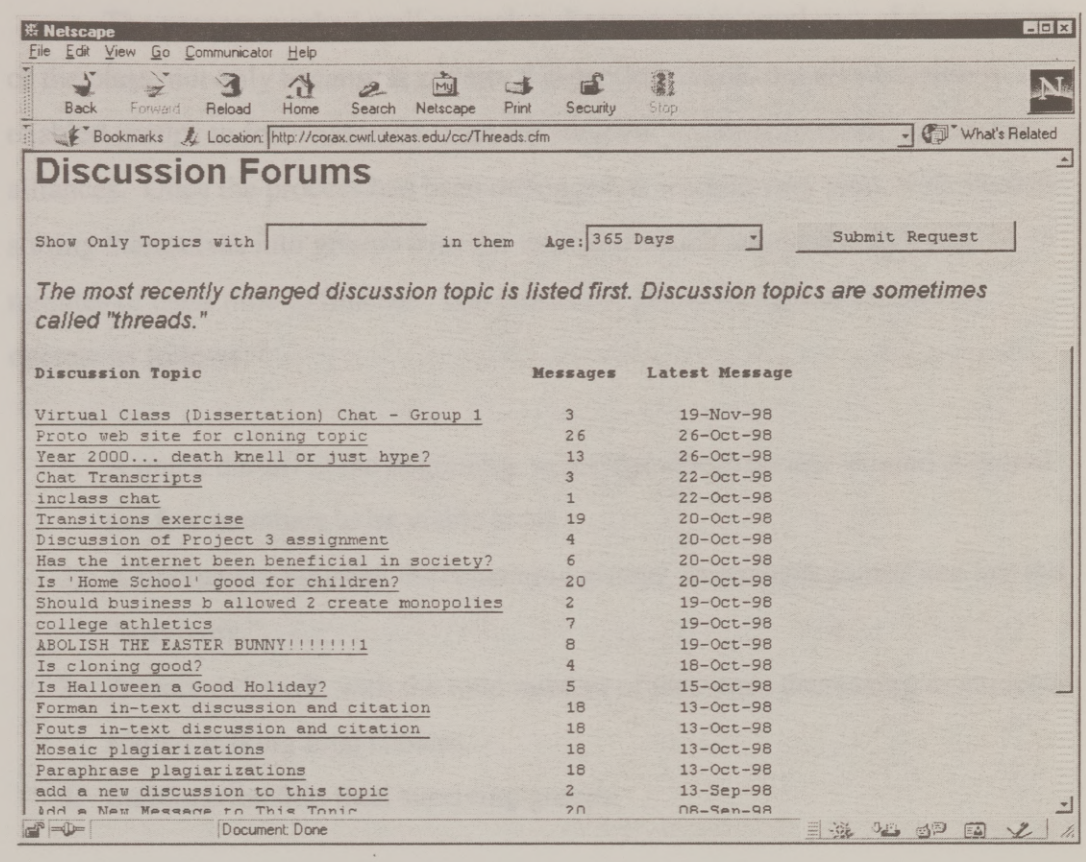


Figure 12: Self-organizing teams in the Forums.

Note how similar these rules are to those which govern cellular automata. They specify the behavior of individuals, but do not dictate what order will emerge.

As with cellular automata, the rules allow groups to shrink as well as expand. The code allows students to withdraw from groups by deleting messages from threads, so that groups can change size dynamically as student interest shifts. Furthermore, the rules also allow students to join multiple groups at the beginning. This allows students to get a quick idea of who is interested in what, without having to commit themselves to any one group in the early stages.

The process worked well enough to become an integral part of the ecology of the class, not only because it governed group formation, but also because it enabled groups to form naturally from the students' existing interests, goals, and alliances. Once the process had been debugged, it worked very well, with students sorting themselves into groups over the weekend which required only a few minutes of class time to finalize. The process of group formation seemed to operate as follows:

1. A quick die-off at the beginning, as groups whom no one wanted acquired too few members to be viable at all.
2. A fluctuation between the remaining groups, as students joined and left the "survivors."
3. A second die-off, with the total number of messages decreasing as students started making final choices.
4. Stabilization into final surviving groups.

The fluidity and dynamism of this process enabled students to assess the forming teams and make a considered decision. Everyone had the opportunity to propose a topic, and everyone had time to consider which group they preferred to join. For these reasons, I would say that this process is *better* than the one it replaced.

In interviews, I learned that some students had made creative social improvisations to make the process serve their wants better. One student told me that she had not waited for a group to form of itself, but had instead recruited members to her topic. Another student commented in an interview that he chose his groups not for the topic, but for the people in them:

> I was looking for certain people to work with, and I was waiting for them to come forward. I guess I was more interested in working with a particular person than in the topic itself.

This same student noted that choosing by people made it risky to propose one's own topic, because one didn't know who would join it. (In fact the rules permitted students to withdraw from their own topics, but students didn't always fully grasp the rules of the game.) This suggests that group-formation was a fluid and complex process, where multiple actors were basing their actions on the concurrent actions of others and carefully monitoring the current state of affairs. This is consistent with the dynamism of an information ecology, where decision-making and participation are distributed throughout the organization rather than centralized.

The Dynamic Syllabus

A key managerial tool is what Peter Drucker (1988) metaphorically calls a "score": a set of "clear, simple, common objectives that translate into particular actions" (p. 9). In music, a score is a set of instructions which are interpreted differently by particular specialists. As Drucker notes, a cellist will interpret the score in one way while the tympanist will interpret it in another, based on their particular skills and imperatives. The job of the conductor is not to tell the musicians how to play their instruments – for that knowledge is what makes them the specialized practitioners they are – but to mediate between them: to help them relate their performances to the group's. Drucker stresses the interpretibility of the score as a means of maximally empowering the knowledge worker and team.

Ikujiro Nonaka (1998) takes interpretibility even further, in urging that directives be deliberately metaphorical and vague in order to evoke creative responses.

The metaphor of the score is useful because it helps reconcile the centrifugal force of locality with the centripetal force of leadership. The score grants autonomy to teams, yet keeps them from becoming completely isolated from each other.

In an academic course, the score is the syllabus. However, unlike Drucker's score, the syllabus is usually not a particularly interpretible or changeable document; it is normally a simple chronological list of readings and due dates. Everyone draws the same imperatives from it and acts in the same way. A traditional syllabus is often rooted in a transmission model of learning; it could be said that it resembles the command-and-control model of management.

However, in a Web application, the syllabus can be more like Drucker's score. When it is generated dynamically from a database, the idea of change is implicit in its structure. (In fact, the Collaboration Center's Syllabus module warns students not to print out the syllabus, since it is potentially such a fluid document.) A dynamic syllabus can be constantly fine-tuned to reflect the needs of the class, responding organically to circumstances as they occur. It could be said that a dynamic syllabus functions as the DNA of a classroom information ecology.

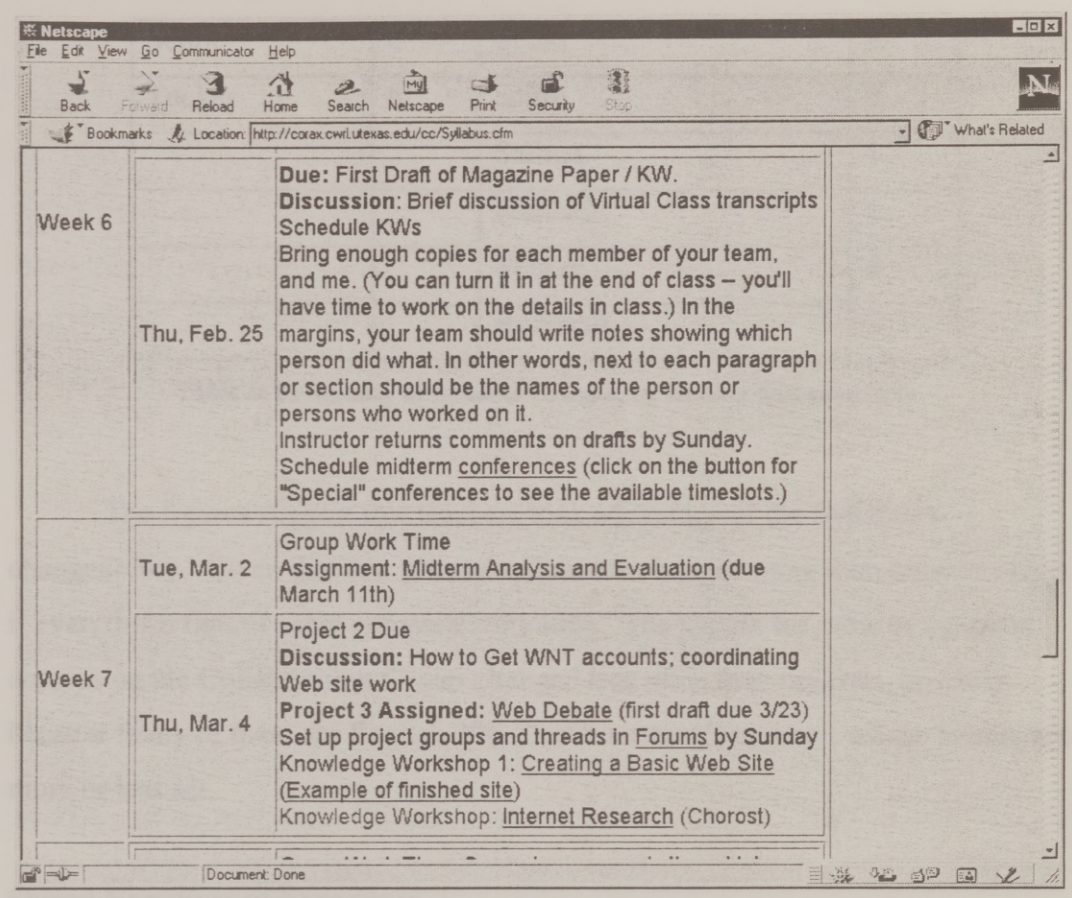


Figure 13: The syllabus module.

The Collaboration Center's teachers changed their syllabi relatively often. In the figures below, I've defined a series of changes in a short time period (>2 hours) as a single editing session.

Fall 1998		Spring 1999 (to 3/7/99)	
Chorost	40	Chorost	22
Rowland	6	Barrish	1
Rowland	5	Mallinick	3

Schilt	6	Schilt	8
Tucker	6	Schilt	6
		Salinas	1
		Kimball	1
		Tucker	4
		Rowland	2

Table 3: Frequency of syllabus changes, by teacher and semester.

The figures suggest that teachers took advantage of the Syllabus's changeability. Given that few printed syllabi are changed more than once or twice, if ever, those figures seem comparatively high. The syllabi for most of the other courses on the Collaboration Center changed less often than my own, possibly because many of them are English 306 (freshman comp) courses, whose syllabi are more or less set.

Shared Narratives of Experience

Crucial to any ecology of knowledge is, of course, knowledge itself. "Knowledge" is not simply know-what, e.g., the facts and rules-of-thumb which accumulate in any organization, but also know-how: the creative skills, social practices, and tacit knowledge which enable people to work effectively and magnify their productive impact (Spohrer, n.d., online).

One of the most effective ways of sharing knowledge is through the telling of stories (Brown & Duguid, 2000; Kleiner & Roth, 1998; O'Dell & Grayson, 1998). Brown and Duguid discuss how one of the most effective training methods at Xerox was the decidedly low-tech lunchroom ritual of telling "war stories" about

difficult copier repairs. Consonant with this is O'Dell and Grayson's observation that the lower-tech the mode of information transfer, the higher-grade the information transferred is likely to be. They observe that data mining, a very high-tech kind of information technology, yields reams of data but comparatively little knowledge. On the other hand, the low-tech technology of telephone help desks and hallway conversations yields high-grade insight. O'Dell and Grayson imply that their observation has the force of natural law: "...The higher the grade of knowledge, the lower-tech the solution. The two are inversely correlated" (88).

These findings suggest that an effective learning environment should have some way to facilitate the sharing of stories. This is easier said than done, because storytelling is by nature an impromptu activity deeply embedded in social and interpersonal contexts. Some efforts have been made to annotate and index videotaped lectures, such as the Madcap project at Xerox PARC, which attaches audience notes and indexes social cues such as laughter and clapping (Brown & Duguid, 2000, 245). However, the telling of stories is such a social event that it remains difficult to record for later use.

Though I certainly can't say I've found the answer to this challenging problem, I have experimented with ways of approaching it. The Collaboration Center's knowledge base module is a repository where texts of any length can be stored and hyperlinked to other articles. At the end of the Fall '98 semester, I asked my students to describe in their final self-evaluations what they thought it took to work successfully with other people. I was impressed by the depth and thoughtfulness of many of their responses, so I put together a compilation of the best ones, and asked my Spring '99 students to read the resulting article. (This was

easily done because, by design, Knowledge Base articles posted in one semester can be read by students in all past and future semesters.)

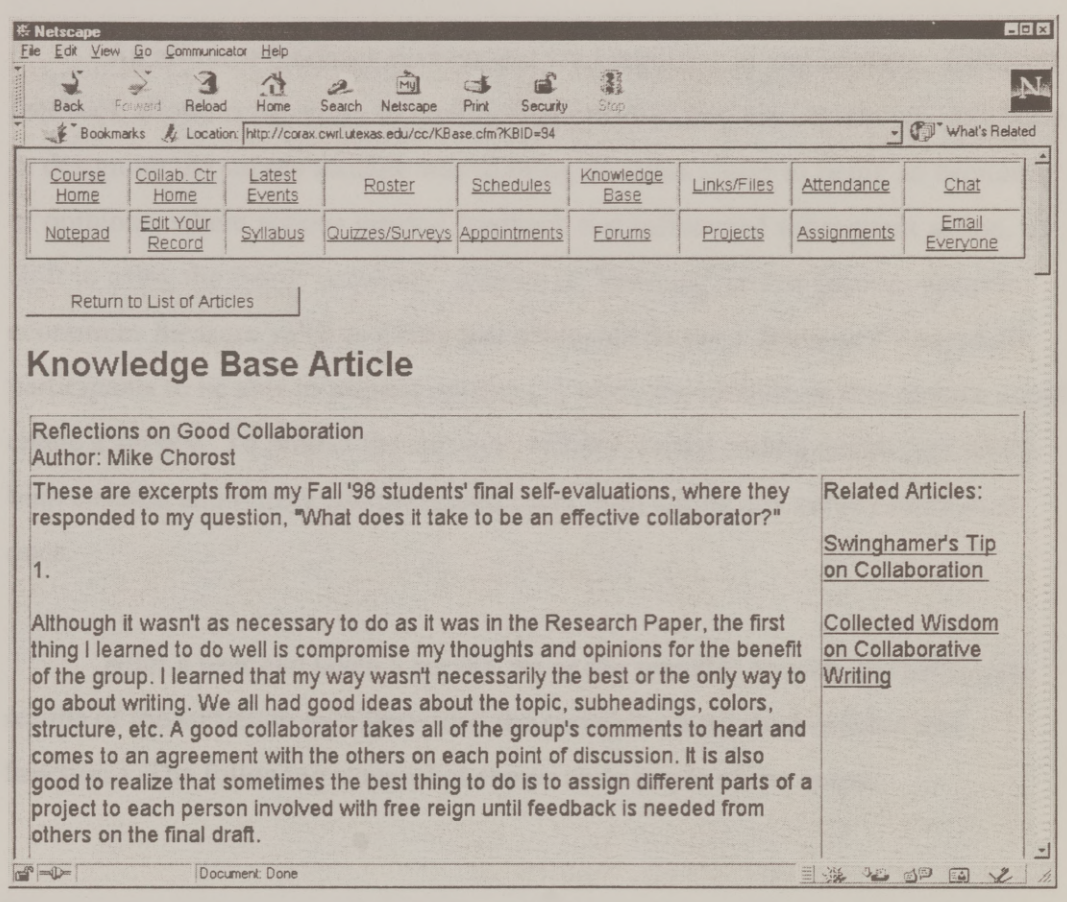


Figure 14: The knowledge base module.

Students seemed to enjoy reading these stories when they were assigned. It would be a worthy project to assess the actual value that these stored stories had, in terms of helping students better anticipate likely challenges in teamwork.

Opinion Economies

An efficient feedback process is a vital regulator in any ecology. Efficient feedback allows ecosystems to rapidly adapt to changing circumstances. My hope in designing the survey module was to bring the class closer to being an economy of opinions, where actions yielded feedback that influenced subsequent action. (I shift to using the word "economy" instead of "ecology" in this section, because economies measure value in a way that ecologies do not.) Because I wanted all participants to be able to request opinions, I wrote the module so that anyone could create a survey. To protect the surveys' validity, I used coding techniques which both ensured anonymity and prevented users from taking any survey more than once.

After a user had taken a survey, he or she was able to view the aggregate results of the survey to date (provided the survey's creator had enabled that function.) The following screenshot shows an example of the output.

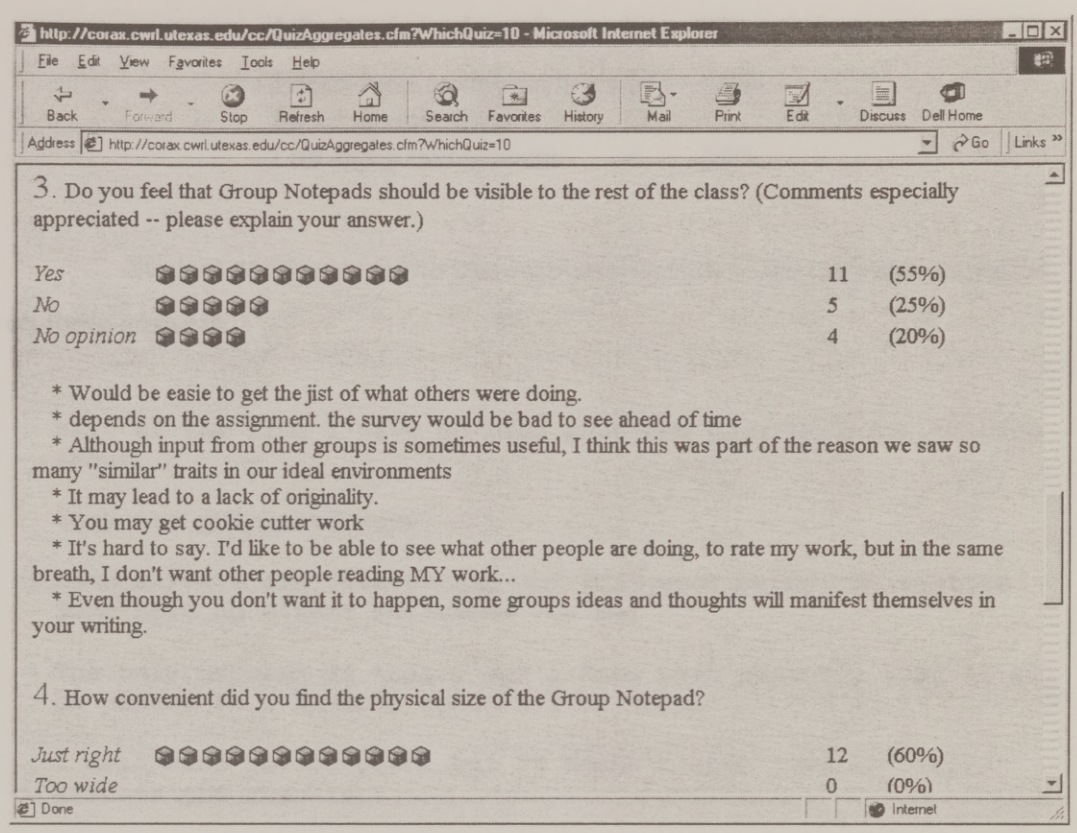


Figure 15: The output of the survey module.

As a teacher, I found the Survey module extremely useful for getting demographic information and feedback from the class. However, students had mixed feelings about it. Perhaps this resulted from overuse; there were 25 surveys posted between January 19th and the end of February 1999. Of that number, ten were mine, and the rest were by students. Ironically enough, the best way to find how students felt about taking those surveys was another survey. The question was, "How do you feel about taking the Surveys? Please comment to amplify your answer."

I don't enjoy taking them	2	9%
I don't have strong feelings for or against them	14	64%
I do enjoy taking them	6	27%

Student comments in the free-response section of the question amplified on these answers.

> half of the time I have no idea what the questions are talking about.
>
> they are quick and easy.
>
> Some surveys are interesting and I'll even study the results. Some hold no direct relevance for me.
>
> The only problem is that I don't know when exactely they'll all be posted so I haven't taken all of them.
>
> I dont mind taking them, but it doens't seem like we really discuss the results.
>
> They tend to get tiring if large numbers are required.
>
> They seem to be usefull tools to see how everyone else in the class feels about a certain subject so taking them is one way of showing gratitude to those who took mine.

The majority of students (64%) indicated neutral rather than positive attitudes. I think the number of positive attitudes would have been higher if there had been fewer surveys: "They tend to get tiring if large numbers are required." Another comment supports my belief that some of the surveys were too poorly constructed to be useful: "Half of the time I have no idea what the questions are talking about."

This last comment points to a major problem with student-written surveys. It quickly became clear that students rarely know how to write survey questions in

methodologically valid ways. As one of the introductory assignments in the course, I had groups of students design a survey and analyze the results. Topics ranged from "Dating Habits" to "Weekend Activity" to "Internet Usage Habits." All of my students' first attempts at surveys were partly or wholly unsuccessful. They generally did not understand how to write questions as neutrally as possible, and to provide a range of multiple-choice answers that yields real insight into the respondents' opinions. Many questions were insufficiently precise ("What are your feelings concerning Internet use?") or made oversimplified assumptions ("How much do you spend, on any given weekend, on alcohol?") Many answer sets failed to cover the full range of possible responses; for example, a survey on dating attitudes neglected to consider that some respondents might be married.

Therefore a class which used this tool would also have to teach a primer on survey methodology. This would fit very well into a writing class: surveys are a different but arguably no less valid kind of writing than creating papers and peer reviews. Survey-writing prompts students to think about issues of audience, methodology, and interpretation. Future work in this area could involve building lessons that guide students through writing valid surveys and evaluating their results. These could perhaps be given in condensed form to students in other fields.

It is also important to integrate the survey results more carefully into life of the class. My follow-up on Surveys left something to be desired: one student commented that "it doesn't seem like we really discuss the results." For a survey to be maximally effective, there has to be some degree of collective follow-up. In the same vein, the last comment ("They seem to be useful…") raises the prospect of reciprocity: "taking them is one way of showing gratitude to those who took mine." It appears that this student regarded taking surveys as a *social* act. It was a way of

providing information as a way of "paying" another student for having done the same earlier. This attitude – if encouraged by the teacher – would help the survey module support a true economy of opinions.

What these four techniques – self-organizing teams, dynamic syllabi, shared narratives of experience, and opinion economies – have in common is an organic, decentralized architecture. In each of them, the outcome is dependent on the multiple inputs of all the actors in the classroom. (The syllabus is somewhat of an exception to this rule, since the teacher controls it, but even here the teacher is influenced by student feedback.) Together, these four tools contributed, in different degrees, to the interactions which constituted my courses' information ecologies.

Constructivist pedagogy calls on teachers to be environment designers and managers. To that end I've shown how I designed, and how students used, many of the functions built into the Collaboration Center. I've discussed tools that supported locality (team rooms, email, group notepads, and the Attendance module), the topologies of human networks (rosters, announcements, and activity tracking), and ecologies of knowledge (self-organizing teams, dynamic syllabi, narrative sharing, and opinion economies.) These tools and usages supported the interactional ecologies of the classes I taught, by facilitating rich multilateral communication, enabling the efficient distribution of information between individuals and groups, and supporting self-organizing processes such as team formation.

Chapter 3: Design Principles for a Next Generation of Collaborative Learning Environments

1. The Agenda of Software Design

The software used in a networked classroom is a major influence in how courses in that space are conducted. It embodies the designers' beliefs about communication, teamwork, privacy, and power, and enforces them by affording certain acts, modifying others, and making others impossible. In using the software, teachers add another layer of influence, by choosing which parts of the software to use, and how, and when, guided by theories of teaching and learning. Numerous other actors – students, administrators, systems administrators, interaction designers, programmers, support technicians, fellow teachers – exert additional influence through their actions, suggestions, and prohibitions. The outcome, how a given course proceeds on a day-to-day basis, is a complex epiphenomenon of all of these influences.

The outcome of a given class's interaction with software is thus by no means solely determined by the software itself. The software is embedded in an environment that is thoroughly social and political. Cynthia Selfe (1992b) and Philip Agre (1996) have both urged teachers to be aware of the "politics of technology," in order to avoid being unknowingly driven by the agendas, both hidden and explicit, of those who build and market hardware and software. They advocate that educators' agendas should be driven by social goals rather than by technological capabilities. Agre writes,

> A technology-driven social agenda posits an inevitable and autonomous line of technological development, from which social consequences flow. A

> socially-driven technology agenda paints a picture of which computers are just one part, and it tells a story that depends simultaneously on nontrivial ideas about how computers work and nontrivial ideas about how society works (Online).

To support a social agenda, Agre suggests that academia needs to develop three concepts: collective cognition, community systems design, and developmental democracy. "Collective cognition" is about developing communication tools to support institutions, such as unions, support groups, and professional associations. "Community systems design" is about engineering the physical and virtual environment to support socially desirable actions: its goals are solidarity, social capital, and "the community's sense that it is taking control over its collective life and fate." "Developmental democracy" is about teaching school-age citizens how to organize events, build consensus, articulate issues, and perform other important civic tasks. Networked computing, Agre suggests, offers new and valuable opportunities for developing a pedagogy which supports these concepts. This pedagogy would use technology, but would not be driven by it.

Cynthia Selfe (1992a) also urges teachers to base technology choices on social and pedagogical needs rather than technological possibilities. Like Agre, she frames the issue in architectural terms when she advises educators to "prepare English teachers to be architects of computer-supported learning spaces and virtual learning spaces" (p. 35). She notes that the space of the English classroom has already been rethought and reconfigured because of the computer – not always effectively – and calls on teachers to continue interrogating fundamental decisions about access privileges, the organization of group and individual work, the construction of discourse situations, and privacy.

Many observers as well as Agre and Selfe have argued that if teachers lack awareness of the sociopolitical dimensions of technology, decisions about its usage may effectively be taken out of their hands by computer manufacturers, software vendors, or school administrators, who may have agendas different from theirs (Sommers, 1992; Zeni, 1994). Paul LeBlanc (1994) noted that in the 1980s and early 1990s many schools' purchasing preferences ran to drill-and-practice and tutorial programs, which served a narrowly vocational pedagogy rather than a constructivist one. He suggested that in this vocational pedagogy,

> Knowledge becomes a set of facts and literacy a series of mechanical and rule-bound acts – in total, the computer as tireless "tester" supports what Ohmann calls the "meager literacy" of "subordinate classes" (p. 31).

Current educational software has moved well beyond drill-and-practice, but even software specifically designed to support a constructivist pedagogy can be used to serve other political ends. Morrison and Goldberg (1996) offer a scenario of how networks, usually considered a constructivist tool *par excellence*, can be used to serve a more restrictive politics:

> Networks can open schools up to the outside world, flatten hierarchical relationships, and distribute responsibilities for teaching and learning, and for managing educational resources. The same technology, put to different uses, may have exactly the opposite effect – tightening central controls, narrowing the scope of teaching and learning, and concentrating management of resources among a small number of designated experts (p. 127).

The first outcome can be called an "agora" outcome, after the open marketplace of ideas, whereas the second can be called a "panopticon" outcome, after the Benthamite plan for a prison with ubiquitous and centralized surveillance. For a constructivist pedagogy, the first outcome is greatly preferable. Yet, as with

physical architecture, an architect's intentions can often be easily defeated. Therefore it is important that teachers develop robust social agendas which help them bend technology to their needs, rather than vice versa.

What could this robust social agenda look like? As I discussed in Chapter 1, boundary management is generally thought of as a teacher or manager's behind-the-scenes planning and one-on-one interventions. But groupware is also a form of boundary management. It is a form of *software* engineering that is also *social* engineering. If humanely designed and implemented, groupware can allow teachers and students to make thoughtful decisions about their working processes and invest them in visible, autonomous entities. Good software design should support the autonomous decisionmaking and work of student teams, facilitate communication between all actors on the educational scene, enable teacher oversight and management, and offer scope for the ready improvisation of new online tools and processes.

2. The Meaning of "Design" and "Use" in a Social Context

Design and Metadesign

Terry Winograd and Fernando Flores (1987) argue that design is always and already an intervention into, and an expansion of, social activity:

> In creating new artifacts, equipment, buildings, and organizational structures, [design] attempts to specify in advance how and where breakdowns will show up in our everyday practices and in the tools we use, opening up new spaces in which we can work and play (p. 163).

From this perspective, design does not so much control activity as it structures possibilities for activity. Terry Winograd's own software, The Coordinator, is an example of this perspective in action. It structures message exchanges within an infrastructure which requires commitments and fulfillments but leaves open the choice of what those commitments and fulfillments will be.

In the following sections, I offer four design principles which afford similar kinds of flexibility. These principles are really *meta*design principles, because they lead toward designs which allow *users* to design. As an example, consider SimCity, a popular computer game. It does not make sense to call the SimCity player a mere "user," because he or she is very much designing the growing city, in making choices about power generation, transportation infratructure, greenspace, taxes, and so forth. The player can even make choices which violate the stated intent of the game, such as creating a city solely for the purpose of enthusiastically inflicting disasters upon it. As Ted Friedman (1993) notes, SimCity is at the far end of a continuum of games which privilege authorial power versus games which privilege player power. Maxis, the original vendor of the game, took pains to call it a "Software Toy" rather than a game. The current advertising for SimCity emphasizes the array of tools players can use to design and control the cities they build.

Designing a game like SimCity is very much about building tools that players can use. By contrast, designing a game like Quake is about creating a user experience. The essential distinction is between designing for "use" and designing for "design." One designs Quake or a word processor to afford use, but one designs SimCity to afford design. On the whole, design for use currently

dominates the educational software industry. In this chapter I argue that the industry needs to shift toward designing for design – that is, metadesign.

Metadesign principles are important because they fit into the constructivist program of enabling teachers and students to design their own learning and working environments. They afford community systems design, the careful engineering of the physical and virtual environment to support socially desirable actions.

Before discussing metadesign in detail, I want to avoid giving the impression that users can only engage in community systems design if designers let them. In reality, users can engineer original environments with tools which were designed with very different ends in mind. For example, Donna Reiss (1999) has creatively used the Web and email to simulate a more full-fledged collaborative environment:

> Because there's so much competition now for that one [electronic] classroom, I cannot book it often enough to use it the way I'd like. Therefore, I have sought alternative means to offer students electronic collaboration: asynchronously with the Web. One of my colleagues told me that I am using the Web and e-mail as if they were Daedalus Interchange (Listserve, March 2, 1999).

It is plausible to say that Reiss *designed* an online collaborative environment by putting together email and the Web along with an elaborate protocol for using them. Similarly, my students and I designed a method of letting teams organize themselves by creatively appropriating software which I had originally designed for very different purposes. These examples suggest that design intent can actually be a relatively minor factor in how software is actually used.

One might call this kind of usage "complex usage," for usage which goes against the grain, so to speak, of the original design intent. Several articles have pointed to creative uses of low-tech tools to support complex activities (Justine K. Brown, 1999; Hickey & Reiss, 1997). Another example is a software experiment where teachers discovered that students in a legislative role-playing simulation were stealing passwords in order to stage "Watergate-style break-ins" on other students' accounts to gain advance information on opposing political parties – certainly not an intended use of the software (Reimer & Edelson, 1999). While simply selecting software modules or configuration parameters could not be called design (Repenning, Iouannidou, & Phillips, 1999), the activity of purposefully embedding software into a larger context of activity can be called, at the least, community systems design.

If teachers and users can do so much by using software *against* its design intent, then they could probably do better with software which is *intended* to let them design creative uses. In effect, such software would amount to being a toolkit for designing groupware.

Social Orientation as a Success Factor in Software Implementation

Claudio Ciborra (1996) breaks down the word "groupware" to show that it encompasses both the social and technical realms:

> The word 'groupware' included two distinct elements: a socio-organizational one: the 'group', a collective way of working, collaboration, the intimacy of staying together and sharing (Hackman, 1990); and a technical one, the 'ware', the artefact and the tool. The term 'groupware' connects the two

worlds, the one of human, collective endeavour, and the artificial one of the artefact" (p. 4).

Likewise, Bullen & Bennett (1990) call groupware "simultaneously a social and technical intervention" (p. 297). The CSCL and CSCW literature demonstrates that groupware works most effectively when it is implemented with a specifically social perspective. Discussions of successful groupware implementations usually reveal that the organization understood the linkages between the "group" and the "ware" and designed them into the total system. The key reason "Zeta"'s implementation of groupware to aid tech support (discussed in Chapter 1) succeeded is because the organization created social mechanisms that rewarded careful documentation, such as building it into performance evaluations and using it in training (Orlikowski, 1996, p. 36). Similarly, experiments with distance education technologies have shown that using them in a community context can be more powerful than traditional face-to-face lecturing situations. Students who watch a videotaped lecture with the ability to pause it to discuss questions and issues in a group have been found to outperform students who actually attended the lecture in person (Smith, Sipusic, & Pannoni, 2000). The success in these cases is due to the integration of the information delivery system with a socially relevant situation where it can be assimilated.

Discussions of failed software implementations usually reveal that the causes are social rather than technological. Schneier (1995) suggests that some applications fail because of resistance or indifference (p. 392), and Hara and Kling (1999) suggest that they can fail through poor support of technical problems and lack of timely instructor assistance. Jakob Bardram (1997) has suggested that groupware applications fail when they do not take into account the unique and idiosyncratic characteristics of the workplace where they are introduced (p. 251).

Markus and Connolly (1990) show that applications fail when they do not acquire a critical mass of users such that the costs of using them are outweighed by their social benefits. Galegher and Kraut (1990) analyze the relative failure of a computer-mediated writing project by explaining that their software "expands group members' access to each other by enabling them to communicate without being co-present, but it also restricts the range of their interaction by imposing delay and narrowing bandwidth" (p. 75). For example, the delays in responding to email frustrated group members, and email also led to "pairwise" communication where two members of a group would reach an agreement without sharing it with the others. These are all examples of failures, not of technology, but of community systems design.

However, technology in itself does impose significant limitations in community systems design, and these have to be identified and considered so that a set of design principles can work around them. In the next section, I'll outline some of the theoretical and practical limitations of software. There are certain things, I'll argue, that software alone cannot do, no matter how carefully it is designed. In the section after that, I'll propose a solution to this problem, which is to *design* to afford *community systems design*. The rest of the chapter will be devoted to proposing and discussing the four design principles I mentioned above.

3. The Limitations of Software

There are two kinds of limitations to the success of groupware: practical and theoretical. Of the two, the latter are the more significant and far-reaching.

Practical Limitations

Despite many technological advances, computers are still very limited in their ability to represent work. Groupware applications tend to limit themselves only to kind of information that can be readily represented in electronic format, resulting in the exclusion of whole realms of experience and representation: hallway conversations, paper documents, customs, phone conversations, and many other kinds of information that bear on work (Davenport, 1997, p. 25). The result is a disproportionate focus on easily representable objects such as files, documents, and textual conversations. In a way, groupware designers are still like the protagonist of an old joke: they are looking for their keys under a streetlamp because that is where the light is.

Other limitations are rooted in the specific characteristics of computer technology. Even large computer displays impose on the users a kind of peripheral blindness, because they can usefully display only one or two documents at a time. Little room is left over for displaying an object's spatial or temporal relationship to to other objects, even when the software capability is there. Various solutions to the "real estate" problem have been offered, some involving immersive three-dimensional space (Novak, 1994) and others wall-size displays (Creighton, 1998), but realistic applications for workplace environments seem far off. Furthermore, the weight, cost, and bulk of even "portable" computers limits where and when they can be used. It could be said that the main problem with groupware is that it has to run on computers.

Theoretical Limitations

The theoretical limitations are much more fundamental. Groupware pursues the same dream that has been expressed in utopian fiction for centuries and shares its inherent flaws. Virtually every utopia in literature, from Plato's *Republic* (380 BC) to Thomas More's *Utopia* (1516) to Charlotte Perkins Gilman's *Herland* (1915) is about creating spaces and processes that channel human action in rational and productive ways. Utopias typically go to great lengths to specify a physical architecture that directs and channels behavior.[6] However, these usually end up being (or are perceived by critics as being) restrictive and repressive. Furthermore, as I described in chapter 1, human populations behave autopoietically: they rarely settle down to an equilibrium norm, which means that they often quickly outmode the structures set up to contain and channel them. At their worst, architectural reifications of group processes can inhibit evolution and refinement, forcing people to find increasingly elaborate ways of working around them (Winograd, 1988, p. 643).

The fundamental problem can be summed up in the following way: Whereas human action is autopoietic, software is algorithmic. Whereas autopoetic systems transform themselves continuously and nonlinearly, often producing chaotic outcomes, algorithms behave linearly, yielding precisely the same output for the same input. Much knowledge is tacit and situated, which means it cannot be formalized and decontextualized (Polanyi, 1997). Algorithmic systems for shaping action are doomed to be limited by the scope of the designer's vision, for no matter how subtle and elaborate they may be, human activity will always be more so. Furthermore, their scales are different. Algorithms are relatively bounded and

---

[6] I taught a course at Duke titled "The Unbearable Orderliness of Utopias" in 1994.

defined, whereas autopoeic processes are unbounded and undefined. Writing an algorithm to guide an autopoetic activity is analogous to building a shed to house a town.

Winograd and Flores suggest that there may be a way to have software that is not restricted to predesigned algorithms, but it seems a formidable challenge – tantamount to achieving strong AI. They suggest that software may eventually display "evolution of structure" (analogous to the idea of neural nets). They suggest hopefully,

> We get a very different perspective on computers if, instead of looking at the problem of programming them, we look at them as plastic structure-determined systems…Many of the characteristics of autopoieic systems will hold for any system whose internal structure can change as a result of perturbations, and computer programs share this quality (p. 102).

Whatever the merits of this approach, it seems well away. For the moment, programmers are essentially limited to designing algorithms with as much flexibility as they can muster. This does little to solve the problem, since human working environments will always exceed the foresight of programmers.

4. Designing to Afford Community Systems Design

While algorithms are fundamentally limited, criticizing software because it has those limitations is like criticizing a hammer for being unable to drive a nail on its own. Hewitt & Scardamalia (1996) note that simply presenting software as a set of "opportunities" is insufficient if a robust pedagogical and social network has not been engineered to support it:

> The challenge, therefore, is not simply to provide opportunities for distributed processes, because students may not even recognize them as such. Rather the challenge is to design situations and tools that invoke a deliberate bias toward shared activity (Online).

The solution, as Hewitt and Scardamalia write, is not to expect software to be equal to all possible circumstances, but rather to engineer a social environment which will amend its insufficiencies through directed and creative improvisation. Skilled leadership can help bridge the gap between software representations of activity and the material world in which they take place. To put it another way, the solution is to conjoin *software* design and *community systems* design. Software design is what is done by vendors: user experience designers, programmers, quality testers, and so forth. If they are conscientious, they will pay plenty of attention to the social context in which the software will be used. However, they cannot always be available to end-users, who will face a greater diversity in situations than the designers can anticipate. This is where community systems design comes in. This is usually done (consciously or not) at the user level, where users creatively design processes and actions given the materials they have at hand.

Thus there is a need for designers to design software which affords community systems design. There are at least three ways in which it can be done. Two of those are at the design level. First, the software can be designed to give human actors considerable latitude for discussion of the work objects represented. Second, it can be designed in a highly modular way, so that users have large opportunities for customization. The third is at the community systems design level: the teacher/manager can construct a total environment where problems, inconsistencies, and exceptions can be handled and resolved. In the next two sections I will discuss examples of software that is grounded in this reciprocity between design and community systems design.

Examples of Design

In this section I will discuss three programs – The Coordinator, NegotiationLens, and ActionWorks Metro – which are grounded in the language/action perspective. According to Bock & Marca (1995), the language/action perspective

> focuses on how language can be used to intentionally and continually create the workplace. It is founded on the notion that language is generative (as opposed to descriptive) in nature, and this quality gives people a new power both to transform the way they work and the way they build technology (p. 44).

Programs designed with this generative perspective lean toward providing tools for collaboratively building things, be they conversations, agreements, or workflow diagrams. The emphasis is on open-ended building, rather than going through specific processes.

The Coordinator

Terry Winograd's "The Coordinator" is an email-like program which stores and generates messages that are records of actions in ongoing conversations. Participants' messages are structured by rubrics such as "Promise," "Counter-offer," "Decline," "Report-completion," and "Acknowledge." Users can ask to make a request, for example, whereupon the program prompts them for pieces of it (to whom, for what, respond-by, complete-by, etc.)[7] Winograd (1988) identifies a

[7] A similar protocol for conversation is given in Fisher and Fisher (1997, p. 182).

number of kinds of conversations, all of them open-ended, that software might be able to facilitate:

- conversation for clarification – coping with breakdowns or potential breakdowns,
- conversation for possibilities – speculation, brainstorming, and
- conversation for orientation – creating a shared background, telling stories (p. 635).

Winograd explains that The Coordinator is explicitly designed to structure rather than control communication:

> The Coordinator has no magic to coerce people to come through with what they promise, but it provides a straightforward structure in which they can review the status of their commitments, alter those commitments they are no longer in condition to fulfill, make new commitments to take care of breakdowns and opportunities appearing in their conversations, and generally be clear (with themselves and others) about the state of their work (p. 634).

An important aspect of The Coordinator's design is that it presumes the existence of an active social life beyond and outside of it. It recognizes, for example, that "breakdowns" happen and offers tools for dealing with them – e.g., the option to make new commitments. Furthermore, the concept of a "commitment" is sufficiently broad to allow a range of activities to fit into it, from ordering a pizza to running a large project. This, then, is an example of design which is focused on the needs of communities.

NegotiationLens

Another program, NegotiationLens, bears a resemblance to The Coordinator, which it acknowledges as an influence (Adelson & Jordan, 1992). It is built on a theory of negotiation as an iterative and cyclical process where the parties verbalize their needs and compare their solutions, seeking common ground. In its emphasis on naming forms of speech ("needs," "resources," "requests," and so on), it "provides a sort of spread sheet for keeping track of the state of the negotiation" (p. 484). It consists of a series of spaces which prompt users through a process of formulating and comparing problem statements, proposing initial solutions, defining underlying needs and resources, and matching needs to resources. If users discover that they disagree about needs and resources, they can weight them to see what matters most to whom.

ActionTechnologies' ActionWorks Metro

ActionTechnologies's ActionWorks Metro is also modeled on Winograd and Flores' theories (ActionTechnologies Web site, 2000). This software enables users to build applications which facilitate an organization's workflow. It is based on the idea that a task consists of four sequential steps:

- Preparation (e.g. a request or offer)
- Negotiation (deciding terms and parameters of the job to be done)
- Performance (fulfilling the request/offer)
- Acceptance (evaluation and feedback)

Any collective action, no matter how small, requires these steps. Complex collaborative actions consist of chains of these steps; a large task could be broken down into a sequence of these four-step cycles, where the opening phase of one cycle would depend on the closing phase of the cycle before it. For example, the task of writing a section of a paper might be preceded by the acceptance of an earlier section of the paper, and followed by a preparation to write the succeeding section. To carry out the task itself, the writer might *prepare* by accepting a request from the team leader, *negotiate* the size and contents of the paragraph, *perform* the task of writing the paragraph, and then receive *feedback* from other team members on its acceptability. Within each of these steps, there might be smaller four-step cycles, such as requesting and receiving research materials from another team member. The Metro software aims to clarify processes like this by providing a software map which makes clear what requests team members make of each other and their ongoing status. An example "process map" created by the software (seen in the figure below) shows some of the actions it represents. One can see that there is an overall task (a new product approval) which is divided into the four steps of preparation, negotiation, performance, and evaluation. Within each step, there are several sub-cycles, represented by square icons showing four blue arrows going around in a circle. The process is thus recursive, in a sense.

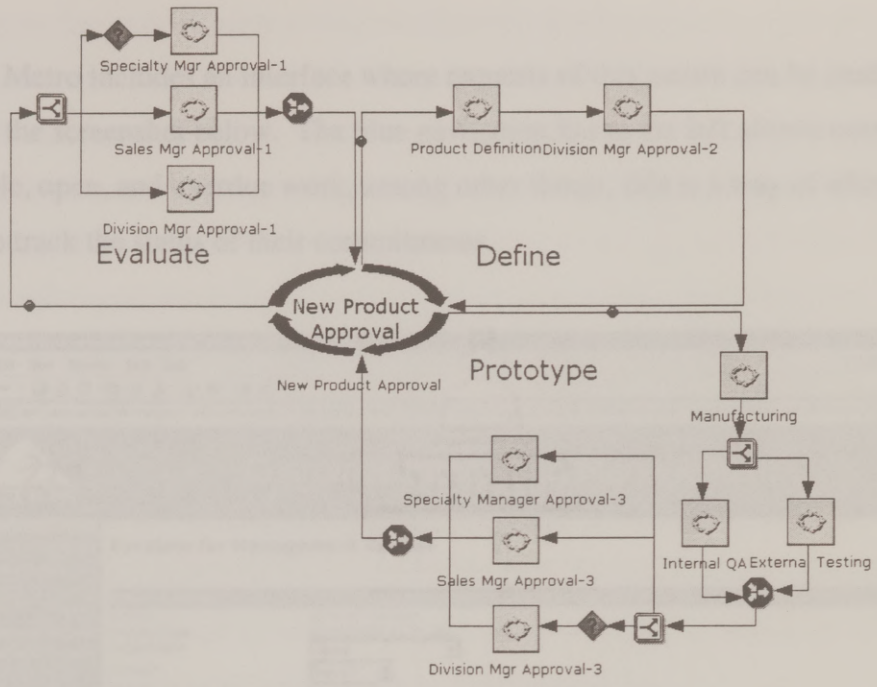


Figure 16. ActionTechnologies' ActionWorks Metro.

ActionWorks Metro was designed for business processes, but it might be adaptable to academic processes. Imagine, for example, a module on the Collaboration Center that allowed students to set up four-step cycles to specify the stages their paper would go through. The actions of each student in the group would be represented as sub-cycles, each with its own process of preparation, negotiation, performance, and acceptance. The ensuing diagram would obviously be complex, with sub-deadlines and various calls for interaction between students, but once created, the software could become the project's conductor, notifying students of what actions are expected of them at any given moment, red-flagging actions left undone, and providing an overall summary of the project's progress. At any moment, a student could log in to ask (in effect), "What do I need to do now?" and "By when does it have to be done?"

Metro includes an interface where requests of this nature can be made, as seen in the screenshot below. The blue navigation bar to the left allows users to see available, open, and overdue work, among other things; this is a way of allowing users to track the status of their commitments.

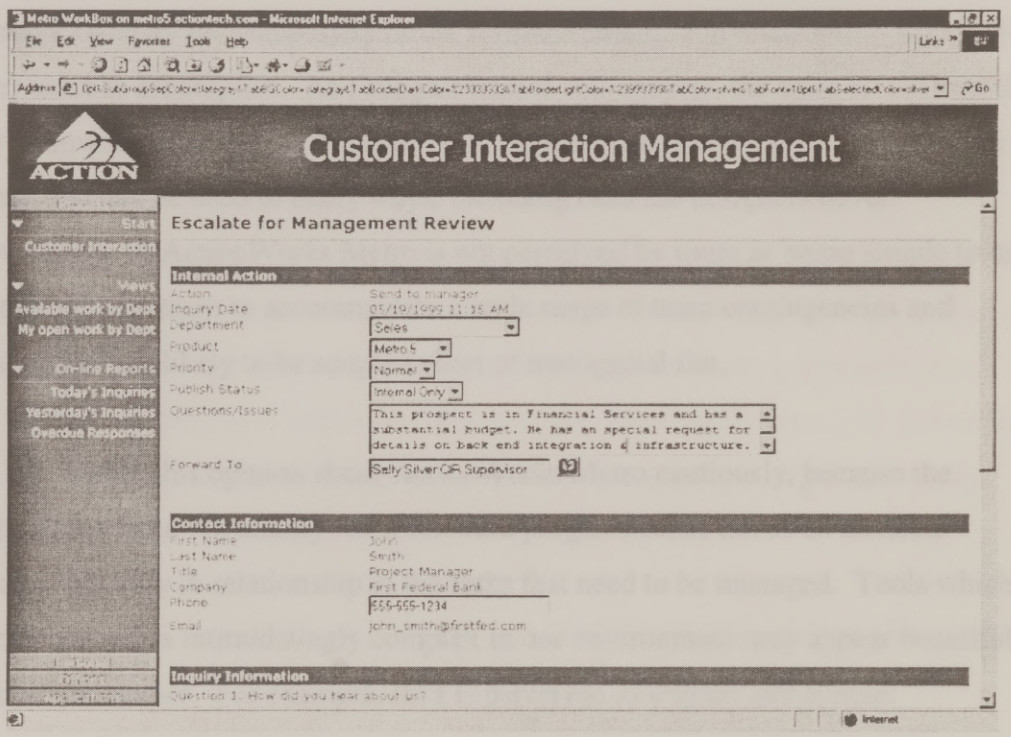


Figure 17: Metro's interface for individual users.

As with The Coordinator, ActionWorks Metro explicitly recognizes the existence of the social life of work, and offers a flexible structure for representing obligations and fulfillments. However, a potential stumbling block with complex representations of the type it permits is that they may be too difficult to update in response to contingencies and changes in plans. As is well known, work often deviates from plans (Grinter, 1997). In order to be effective, software of Metro's

complexity has to be flexible enough to allow for quick revisions of the plan, *and* its users have to have the ethic that departures from the plan need to be instantiated in the plan as it exists online. Given that users are rarely so conscientious as to thoroughly document changes in plans, there may be a substantial gap between the complexity of the software and the practical needs of groups, at least ones which are undertaking relatively small projects. If the gap is large, then users may feel straitjacketed by the complexity of the software structure in relationship to the tasks they want to accomplish. As I suggest later in this chapter, the most effective tools for teamwork have the attributes of both simplicity and flexibility. They are easy to grasp and can be used in many ways, including ones the designers never envisioned. If ActionWorks Metro is not perceived by users as being simple to use and flexible enough to accommodate a wide range of team contingencies and needs, it is not likely to be adopted short of managerial fiat.

I offer this opinion about ActionWorks Metro cautiously, because the "simplicity" and "flexibility" of a software program exists not as an absolute quality but only in relationship to the tasks that need to be managed. Tools which are perceived as intimidatingly complex in one environment may appear beautifully simple in another. For example, Java is perceived as elegant and simple, conceptually speaking, by Java programmers, but is an unwieldy and cumbersome tool for people wanting only to do simple scripting. Similarly, the ActionWorks Metro interface may be too complex for student project work, yet be appropriately suited for a large engineering project.

Whatever their scope, tools for teamwork have to allow their users sufficient flexibility to accommodate their work practices. They have to allow for the inevitable variations, improvisations, and contingencies which arise in any kind

of collaborative work. This is just another way of saying that they have to allow users to engineer practices within them to support desired social goals; in short, they have to support community systems design to a greater or lesser extent. In the next section I will discuss software which I regard as having been successful in this regard.

5. Examples of Affording Community Systems Design

SenseMaker

Software efforts will rarely be successful without teacher/managers who are willing to frame and support them in the context of a community's social needs. A program developed at UC Berkeley, SenseMaker, has been described from this viewpoint. Created to scaffold middle school students' engagement with scientific thinking, it offers a way to group hypotheses and supporting evidence in "claim frames" on the screen (Bell, 1997).

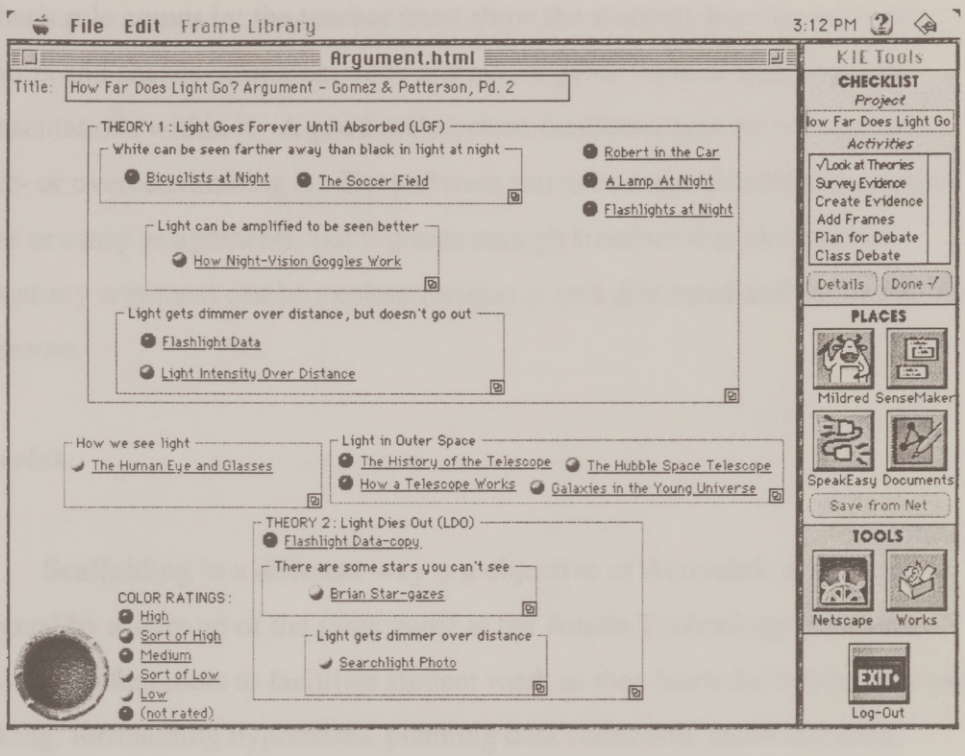


Figure 18: Phillip Bell's SenseMaker.

The screen design represents discourse in spatial terms, by presenting protocols that shape where and when utterances are made. The frames are hierarchical, so that students can create an overall hypothesis and break it down into smaller hypotheses; the program provides spaces where supporting or disconfirming evidence can be presented. Students can indicate their degree of confidence in hypotheses by attaching "evidence dots" of varying brightnesses to them.

SenseMaker's innovation is it that it provides an architecture that scaffolds learning without trying to wholly represent it. Bell notes that students' representations "can only be taken as a shadow of their actual understanding" but believes that they still "support and shape their reasoning" (p. 3). Here is where the

teacher's role comes in: the teacher must show the students *how* the software supports and shapes reasoning, and continue to guide them as they work on representations within it. A good architecture facilitates activity without either under- or overconstraining it. The software provides more direction than paper forms or essay assignments, but it grants enough freedom that any kind of evidentiary argument can be explored within it, and discussed and refined in the classroom.

ActiveInk

Scaffolding in a different way is a objective of ActiveInk, a program designed by a start-up of the same name at the Austin Technology Incubator. ActiveInk offers tools to facilitate student work as they learn the basics of scientific thinking: formulating hypotheses, planning data collection, recording data, analyzing it, and using the findings as a basis for recommendations. For example, one aspect of one program has students design a plan for monitoring litter patterns around their school and then recommending more effective ways to keep the grounds clean. The screenshot below shows a tool which enable students to build an online representation of their school and its litter patterns, which they can then use to record patterns and observe trends over time.

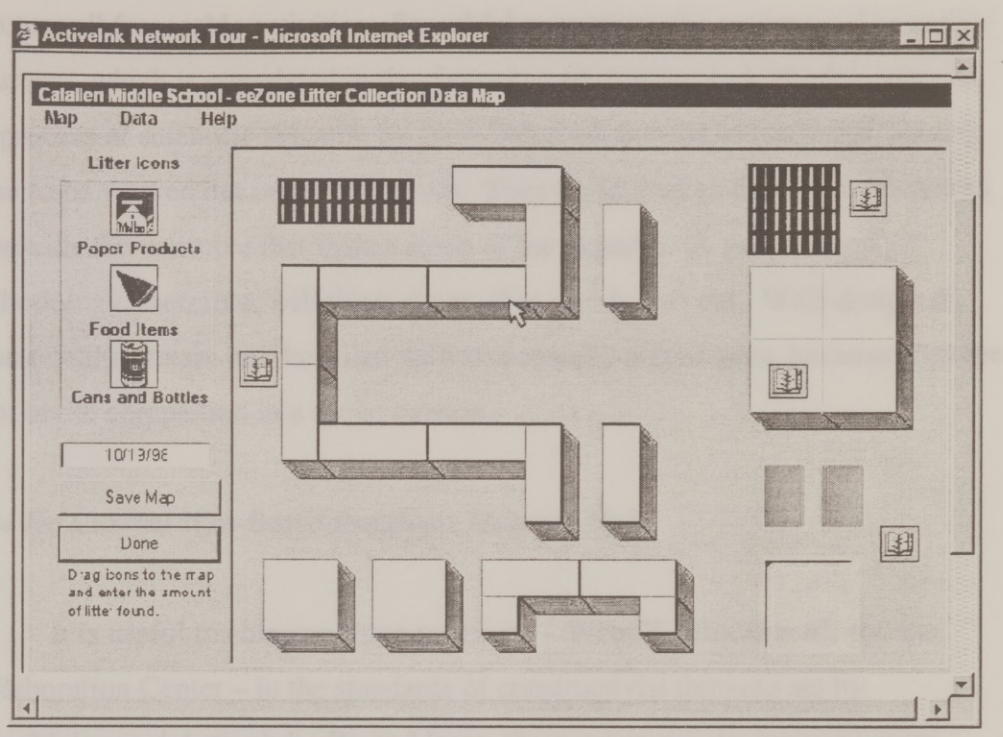


Figure 19: ActiveInk's tool for representing litter on school grounds.

This tool does not march students through a step-by-step research process; rather, it offers a collection of resources to be used as teachers and students choose for carrying out the task of data collection and analysis. One implication of this is that it is possible for students to make poor choices. The idea is that students should be *able* to make mistakes so that they can learn from them. As Eric Jansson (one of its designers) notes, ActiveInk's methodology suggests that a tool that cannot be misused is not a tool worth using (personal communication, December 12, 1999). The program places the locus of control in the classroom group rather than in the software.

These examples suggest that one way to afford community systems design is to abandon the idea of making a design "complete," in the sense of being able to

structure all forseeable activities. SenseMaker supports the outlining of scientific argument, which is completed in the classroom environment; ActiveInk supports the process of scientific research by providing tools instead of roadmaps, leaving those to be worked out in the classroom. They leave it up to the classroom actors to provide the narrative that makes sense of the material. A good design methodology, therefore, will think about what is to be left out. Well-designed educational software will be filled with strategically placed gaps, because gaps are what invite completion in a social context.

How Do Current Web-Based Programs Measure Up?

It is useful to compare three programs – WebCT, Blackboard, and the Collaboration Center – to the standards of constructivist thinking set by SenseMaker and ActiveInk. By and large, these programs are designed more toward management of classroom activity than toward scaffolding the collective creation of knowledge. While they do some managerial tasks well, they still do relatively little to afford community systems design.

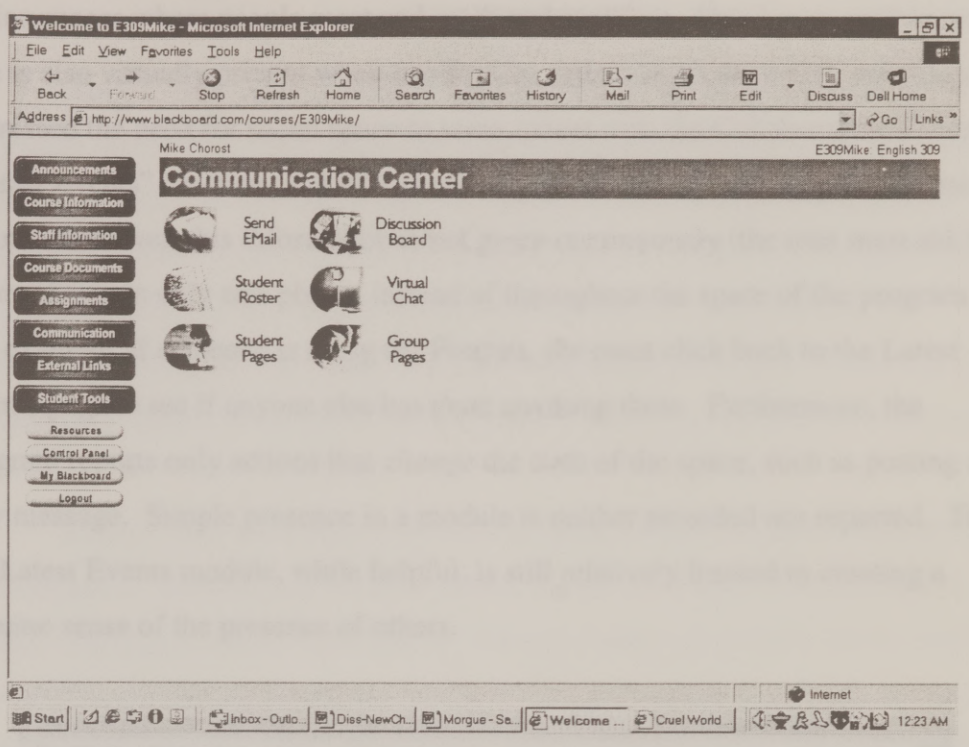


Figure 20: Blackboard.

The functional similarities between WebCT, Blackboard, and my own effort, the Collaboration Center, suggest that there is considerable convergent thinking on what classroom management software should do. Each of these programs permits the maintenance of rosters, syllabi, team rooms, quizzes, announcement spaces, and bulletin boards. It is almost possible to say there is a certain standard inventory of tools which virtually everyone agrees should be available. This standard inventory is generally based on a traditional information systems model of programming, where users are seen as inputting information which is then selectively made available to other users. While such tools are indispensable – virtually any course needs rosters and syllabi – they also inherit the information systems model's limitations. Fundamentally, they create *data* spaces rather than *social* spaces. WebCT and Blackboard are collections of tools, not

online spaces where people meet and work and socialize. One has no inkling who else is also virtually present when one logs in, and no sense what they are doing. I addressed the need for social space to some extent with the Collaboration Center's "Latest Events" function, and with the listing, given on the entry page, of recent logins. However, this information is not given continuously (the user must ask for it) and exists in only two places, instead of throughout the space of the program. For example, if someone is using the Forums, she must click back to the Latest Events page to see if anyone else has done anything there. Furthermore, the program reports only actions that *change* the state of the space, such as posting a new message. Simple presence in a module is neither recorded nor reported. Thus the Latest Events module, while helpful, is still relatively limited in creating a genuine sense of the presence of others.

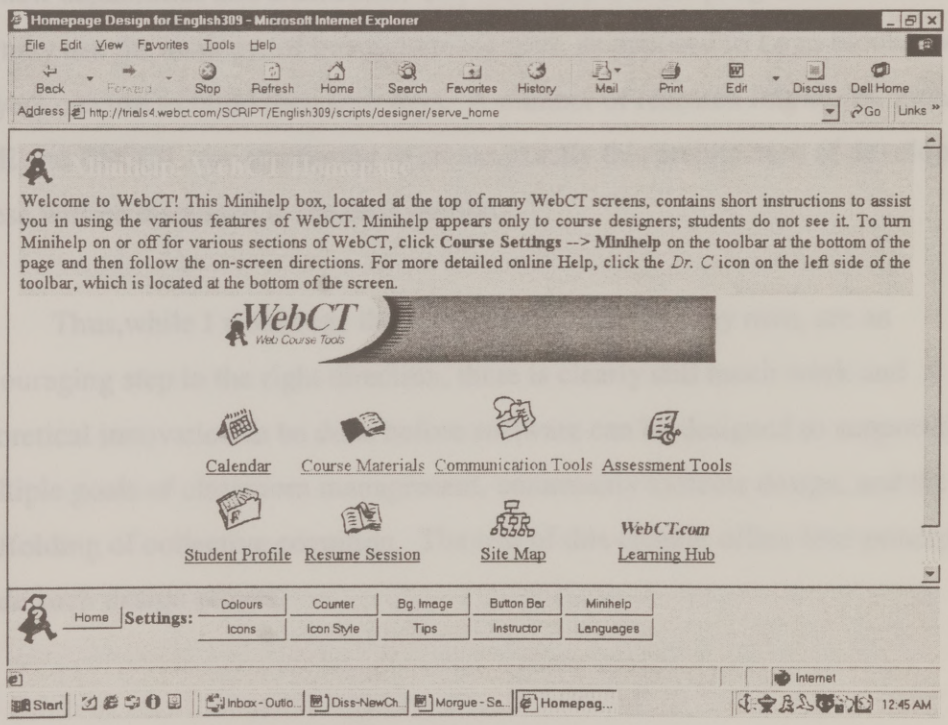


Figure 21: WebCT.

WebCT, Blackboard, and the Collaboration Center have at least two other important limitations. First, their configurability is quite limited. Instructors are essentially limited to turning on and off various features, along with cosmetic options about icons and screen color. It is not possible to use them as building blocks, as it were, to truly *design* an online environment. Furthermore, the positioning of tools on the screen, and their size, remain invariant. A more truly configurable design is the portal design, which I will discuss in the next section.

Second, they are a fixed collection of tools, rather than an architecture of development. Students and teachers cannot design new tools. For that matter, neither can independent programmers. All users remain dependent on the vendor for new capabilities and extensions. For that reason, these programs have the feel, if I may use the analogy, of being premade toys, as opposed to Lego blocks, which players can use to build their *own* toys. A number of research initiatives, including EOE and ESCOT, are attempting to create exactly this architecture of development. These will be discussed in the next section.

Thus,while I would say these programs, including my own, are an encouraging step in the right direction, there is clearly still much work and theoretical innovation to be done before software can be designed to support the multiple goals of classroom management, community systems design, and the scaffolding of collective cognition. The rest of this chapter offers four principles to guide such design efforts.

6. Four Design Principles for a New Generation of Groupware

By putting tools of tremendous managerial power into the hands of both teachers and students, networked computing furthers the constructivist enterprise by making information ecologies ever denser and richer. Networked computing increases classroom actors' ability to communicate with each other, to consolidate the materials of their work, to oversee and manage patterns of collective activity, and to extend their activity beyond the boundaries of the physical classroom. This increased richness and density calls for ever-smarter strategies of design, to help teachers manage the complexity of constructivist education.

The four design principles I propose are specifically chosen because of their applicability to information ecologies. They are:

- Simplicity/Flexibility
- Spatiality
- Modularity
- Extensibility

These principles connect to the idea of an information ecology at multiple points. *Simplicity* paired with *flexibility* recalls the concept of iterating repeatedly upon simple rules. *Spatiality* resonates with the idea of topology, where actors interact in a rich network. *Modularity* recalls the idea of locality, e.g., the relative self-containedness of actors in an ecology. And *extensibility* recalls the notion of biological diversity and evolution: the development of new organisms, behaviors, and niches in response to environmental contingencies.

Simplicity/Flexibility

SenseMaker and ActiveInk's examples of design flexibility lead to a kind of theorem of software design: simple and flexible tools afford complex behaviors. In Chapter 1, I identified locality as one key characteristic of an information ecology, where relatively simple units act on their immediate neighbors to produce global patterns of considerable complexity. The quintessential example of this is the Web itself, where two basic tools – the page and the link – triggered an explosion of complex behaviors and artifacts. Another example is "blogging," a social practice afforded by a simple tool which people can use to build Web-based diaries and link them to other diaries made with that tool. Many enthusiastic web-based communities have sprung up using the tool to link their members together (Mead, 2000). Other examples include classic childhood toys such as Legos and games such as SimCity. In all of these examples, a small set of conceptually simple primitives can be combined to produce a nearly infinite range of outcomes.

Simplicity *and* flexibility must both be present in a tool's design if it is to afford complex behavior in daily practice. This can be readily seen when considering tools that have only one of those qualities. A can opener is a simple tool, but it is not flexible; it is too specialized to afford a range of actions. On the other hand, programming languages are enormously flexible, but they are not simple, and this keeps them from being used in daily practice (except by a small number of elite professionals). However, a hammer combined with lumber and a box of nails meets both criteria, because it is simple to use and affords a wide range of outcomes, from an abstract sculpture to a three-bedroom house. It is both simple and flexible. The principles of simplicity and flexibility help explain why the Web, email, Legos, and other tools have enjoyed such success and popularity.

A "simple" tool can be *very* simple. It need not even seem like a tool at all. Sandy Stone (1995) explores the complexity of the interactions which arise when the telephone is used as an instrument of s(t)imulation by telephone sex workers:

> The sex workers took an extremely complex, highly detailed set of behaviors, translated them into a single sense modality, then further boiled them down to a series of highly compressed tokens. They then squirted those tokens down a voice-grade phone line. At the other end of the line the recipient of all this effort added boiling water, so to speak, and reconstituted the tokens into a fully detailed set of images and interactions in multiple sensory modes (p. 7).

As Stone notes, all it took to evoke a highly complex, interactive experience was the human voice and a telephone line. Likewise, Morningstar & Farmer (1991) found that extremely complex social interactions could be supported using 300-baud connections, a crude graphical interface, and simple tools for interaction. They built an online community called "Habitat" in which users communicated with each other via avatars in a two-dimensional graphic space.

Figure 22: A screenshot of "Habitat."

The users had available to them a set of tools with which they could make buildings and objects, much as in a MOO. These included objects such as paper, glue, and doors for building, and signs, books, and weapons for interacting. From simple primitives like these, complex behaviors such as town-building, murder, theft, and conspiracy arose. Morningstar and Farmer note that the complexity came from the players' actions, not from the designers: "...the complexity...is the product of the users themselves, rather than the system designers – the operators of the system do not have to create all this material" (Online).

Simple and flexible tools have enjoyed great success when deployed in learning environments, because students can easily use them to construct shareable representations of discourse and knowledge. Janet Murray (1997) explicitly connects the principle of simplicity/flexibility with the constructivist enterprise in her discussion of MUDs:

> The student's ingenuity [in using a MUD as a social forum] is typical of the MUD culture. He was taking the materials at hand and repurposing them for his own uses. The notion of reassembling a fixed set of materials into new expressive form was inherent in the original *Zork*, the ancestor of the MUDs, which provided the interactor with a large vocabulary of commands and a rich array of objects that could be combined in multiple ways...This constructivist pleasure is the highest form of narrative agency the medium allows, the ability to build things that display autonomous behavior (pp. 148-9).

MUDs are an excellent example of a simple/flexible tool because they offer considerable scope for creativity and invention, both on the level of design and interaction (see also Dibbell, 1996). For instance, C. Robert Stevens, an instructor

at the CWRL, took advantage of the protean character of MUDs to design sophisticated participatory learning environments:

> … I planned on allowing the user to discover the story in bits and pieces through the reminiscences of those still "alive", newspaper articles, clues left behind on the sites where the action took place in the story, and some supernatural devices such as a disconnected phone that rang whenever someone entered the room and allowed them to listen to a panicked call from the original story wherein the caller's house was crushed by the "horror"…My instructional plan for the space was for students to explore the space, write what they thought the story was in their own words, and then have them read Lovecraft's story to get a sense of how a story can be told in a totally different and non-linear manner, and for how real stories get reduced and compacted when they are set down on paper (personal email communication, February 26, 1999).

Another example of successful simplicity and flexibility of design is a Web-based program called CoWeb. Like Blogger, it consists mainly of the ability to view pages and create new pages linked to them on the Web. Guzdial, Realff, Ludovice, Morley, Kerce, Lyons, & Sukel (1991) observe that the "response from teachers has been to invent a wide range of activities, with the same simple tool" (p. 211). They describe a range of creative uses, from discussion spaces to case libraries to collaborative writing.

Theoretical support for simplicity/flexibility as an effective affordance of complex behavior has been provided by Kvan, Yip, & Vera (1999) and Hutchins (1995). Kvan, Yip, & Vera found that test subjects using high-bandwidth, high information-content communications tools (streaming video, etc.) to collaboratively solve a drawing problem with remotely placed peers spent much of the available time deciding how to represent the objects at hand, to the point of neglecting to actually solve the problem itself. On the other hand, subjects who only had a chat-

style tool available to them tended to solve the problem faster. Kvan, Yip, & Vera speculate that the low bandwidth facilitated a more deeply conceptual approach to the problem:

> Participants in the text only condition were required to focus on the content of the text as the idea emerged, then transfer their to focus to the representation of the idea in text too. As a result, the ideas were explored more consistently at the higher level (p. 331).

The authors also suggest that the text-only participants were required to explore the problem space "more inventively" (p. 331) as a result of the low bandwidth. The simplicity of the text-only environment encouraged more, not less, complex behavior.

Edwin Hutchins's models of group dynamics reinforce Kvan, Yip, & Vera's findings. His connectionist models suggest that complex, multimodal environments do not necessarily lead to greater productivity. The sensory richness of the medium can lead, strikingly enough, to conceptual shallowness. For example, it can rush teams into confirmation bias, the tendency to home in on one conclusion at the expense of considering others. On the other hand, a strategic *reduction* of bandwidth can actually *increase* a group's productivity. "Diversity of interpretations is fairly easy to produce as long as communication among the members is not too rich," Hutchins claims (p. 255).

From these observations it can be conjectured that tools which are simple and flexible afford complex behavior precisely because of their limitations. Simple and flexible tools encourage creative improvisation – in short, play. They allow users to project their own identities upon them. Scott McCloud (1993) writes in *Understanding Comics*, "And soon we discover that objects of the physical world

can also cross over – and possess identities of their own. Or, as our extensions – begin to glow – with the life – we lend to them" (p. 40). In the same way, the simplicity of software tools enables users to make them extensions of themselves. Furthermore, since they can support only subsets of interactive practice, social improvisations and conventions come into play to maximize their limited resources. These improvisations, drawing from millennia of established communicative practices, are highly capable of creating a sense of shared, social space.

This point, about extensibility and the flexibility of software when it is used in a creative, improvisational context, brings out a subtle corollary. While social needs must always be emphasized above technological possibilities when designing software, there is also something to be said for sheer technological play – that is, building tools not because they meet known social needs but simply because they are possible. Tools developed in this spirit may open up possibilities of practice which would never have been envisioned from a standpoint of community systems design. As I noted in chapter 2, I initially wrote tracking functions simply because they were technologically interesting and feasible, and over time developed ways to use them powerfully and humanely in the classroom. In the long run, then, it may be counterproductive to design software working solely from a community's known desires and needs. Tools whose uses have been specified in advance may turn out to be too limiting for the real needs of learning environments.

Spatiality

It has often been noted that virtual spaces offer opportunities for the positive structuring of behavior (Heim, 1991; Benedikt, 1991). Stephen Johnson (1997) writes, "Organized space implies not just a personal value system – as in the

religious order of the Gothic cathedrals – but also a type of community" (p. 62). The way a space is organized exerts both obvious and subtle effects on behavior. In the material world, interaction happens in spaces specifically designed for it – football stadiums, courtrooms, classrooms, and so on. These spaces set up people's bodies and expectations so that interaction generally takes place in well-defined ways.

The concept of space is critical to groupware, because it allows users to perceive it as a *place* where they meet and work. Space comes into being when the designer introduces a time axis, because then the space can be said to exist whether anyone is in it or not (Benedikt, 1991). This is what makes it "real" in functionally same way a physical place is. The concept of a time axis helps explain why email, which is otherwise a powerful communicative tool, is generally not perceived as groupware or a site of work. Like a telephone conversation, an email exchange has no locus in space, and thus offers no way of representing work that is persistent over time and accessible to groups. The same can be said for most chat environments; text-only media become spatialized only when they are connected to databases which can store persistent representations of discourse and ongoing work. This is precisely what happens in MOOs, and also in visually augmented MOOs such as CVW and PAVE.

The requirement of spatiality has nothing to do with the graphical properties of a program's interface; a MOO, which is usually entirely textual, is perceived by its users as being a space. Thus the concept of "space" requires neither a graphical user interface nor a literal representation of material space. Rather, it refers to conceptual and social space. Janet Murray (1997) writes,

> The new digital environments are characterized by their power to represent navigable space. Linear media such as books and films can portray space, either by verbal description or image, but only digital environments can present space that we can move through…We know that we are in a particular location because when we enter a keyboard or mouse command the (text or graphic) screen display changes appropriately. We can verify the relation of one virtual space to another by retracing our steps (pp. 79-80).

The verifiability and objectivity of this kind of space (the ability to retrace one's steps) is what makes it shareable; different users will have essentially the same experience of it.

As with physical space, the objectivity of virtual space permits a powerful consolidation of shared materials. A common way of metaphorizing shared team space is to have "persistent virtual rooms" for group work, stocked with an array of tools such as chat, file storage, and whiteboarding. One user commented of Lotus Notes' TeamRoom software, "It's really cool to have this computer program to help you do your job. It's like a little guardian who tells you what you need to do and has everything right there for you" (Marshak, 1996, p. 197). A student using Myron Tuman's Connect.net (now Norton Connect) has said almost the same thing: "I love the idea of having all of our materials on one disk and that is all we need for class. It is such a gift to scatter-brained people like myself" (Norton Connect website, 1999). This consolidation is one of the ways in which online spatiality is most helpful.

Virtual spaces do not need to offer even a conceptual representation of the material world, since doing so can simply reproduce its limitations.[8] In his

---

[8] The problem with materiality is that it is difficult to reconfigure -- precisely the objection to linear texts articulated by theorists of hypertext (Bolter, 1991; Landow, 1997). Just as in the real world,

criticism of Microsoft's "Bob" interface, which offered a lifelike simulation of a home office, Steven Johnson (1997) writes, "The real magic of graphic computers derives from the fact that they're not tied to the old, analog world of objects. They can mimic much of that world, of course, but they're also capable of adopting new identities and performing new tasks that have no real-world equivalent whatsoever" (p. 61). Douglas Engelbart (1984) takes a similarly anti-literalist approach in his attack on WYSIWYG word processors, which display a direct analogue of the material document being written. He argues that the literalness of the interface is more of a handicap than a benefit, because it privileges the very models of representation that computers were originally designed to overcome. In privileging formatting (material space) over the writing process (logical space), the WYSIWYG interface deprives writers of the ability to create flexible and powerful models of their work.

Spatiality also allows online environments to serve as places of social connection and play. Once space comes into existence, then so do roles. A space can have a creator, objects, and participants (Spellman, Mosier, Deus, & Carlson 1997, p. 198). The members of the group can be informed who is there at the moment ("presence" being a kind of role) and who is not. Their software, Collaborative Virtual Workspace (CVW), is an example of software that goes further to create a sense of online presence than Blackboard, WebCT, or the Collaboration Center do. It offers a virtual floor plan where users can congregate,

---

spatially distributed information is hard to cross-reference, and there are few cues as to the significance of particular items of information. Various theorists have attempted to reconcile the disjunction between physical space and logical organization by imagining spaces where objects constantly reconfigure themselves to represent knowledge in desired ways (Novak 1991). This approach is intriguing, but it still begs the question of exactly how to represent knowledge spatially, and how to train users to manage the complex visual displays that result. The deeper problem is that much knowledge is inextricably contextual, and may not readily lend itself to a naiively direct form of spatial representation.

along with pictures and the easy availability of real-time chat. One gains a direct, visceral sense of real-time activity and can make immediate connections with other online users.

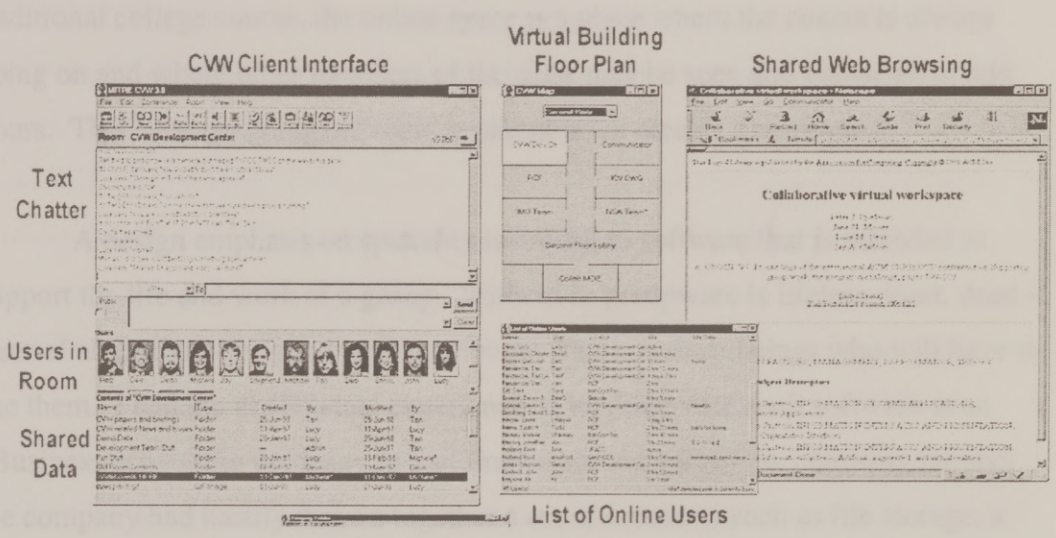


Figure 23: Collaborative Virtual Workspace.

When spaces like these are created, so is the possibility of social life. Paul Cole and Eunice Johnson (1996) have documented how Lotus TeamRoom provided an intangible sense of the "psychological presence" of the others in the group:

> Another benefit of TeamRoom could be described as team building, but is more truly about meeting people's needs for psychological connection and contact…The sense of the psychological presence of others while a person is working alone greatly influences team effectiveness and is frustratingly difficult to achieve. Team Room seems to be heading in the right direction. "I have a sense that we're all together," stated one team member, while another said that, "At your desk you know there's a group." This feeling facilitates the sharing of ideas: "In e-mail I won't send out some ideas to a

wide audience, yet I'm comfortable putting them in Team Room and seeing if people want to read them" (p. 37).

When achieved, this sense of connection can be a powerful aid to groups, particularly if they rarely meet in person. Even when they do, such as in a traditional college course, the online space is a place where the course is always going on and where other members of the class may be seen and contacted at odd hours. This "always going on"-ness is part of what creates a community.

A design emphasis on spatiality is crucial to software that is intended to support the life and work of a group. Without it, groupware is lifeless, inert, dead – a set of algorithms designed with no thought for the human beings who will have to use them. I had the unfortunate experience of working with such software at an eBusiness consulting company in San Francisco. As part of the company intranet, the company had hastily thrown together a set of functions such as file storage, a team roster, an events calendar, and so on, into a screen design which listed them horizontally in apparently random order. There were no provisions for showing users what happened recently, no tools for supporting the social life of the team, and no cues for determining the relative quantity or relevancy of the material in each section. The effect was like inhabiting a cardboard box for wine bottles – an undifferentiated and unfriendly set of receptacles for placing data. No attempt had been made to determine the needs of actual users, and not surprisingly, actual users stayed away in droves.[9] An understanding of software as a spatial entity is essential for creating online environments which will support rich interaction.

---

[9] Shortly after I left the company, the team-support functions were substantially redesigned. I have not seen this redesign and so cannot comment on it.

Modularity

It would go against a constructivist methodology to propose one specific design of academic groupware. Every learning environment is different, and every teacher has his or her own pedagogical goals and methods. The discipline of Computer-Supported Collaborative Learning (CSCL) has acknowledged this situational diversity with calls for modularized software components (Repenning, Iouannidou, & Phillips, 1999; Spohrer, 1999). In this section, I'll propose not a specific design, but rather an architecture of modularity for groupware. The following description is inspired by Plumtree's portal software (http://www.plumtree.com).

Its key elements would be modularity and extensibility. A modular architecture would consist of a number of relativey low-level building blocks which could be snapped together like Legos, each operating independently yet capable of drawing from the same centralized architecture and data. For example, there could be modules for rosters, surveys, syllabi, file libraries, chat rooms, event logs, and so on. In Plumtree's architecture, these are called "gadgets."[10] Visually, gadgets are modules which take up rectangular areas on the screen and can be reorganized, resized, and minimized much as if they were separate Windows elements. Each of the functional areas in the screenshot below is an example of a gadget.

[10] Functionally, the Collaboration Center already consists of modules like this, but they are not architecturally independent of each other: they cannot be encapsulated and moved into new contexts.

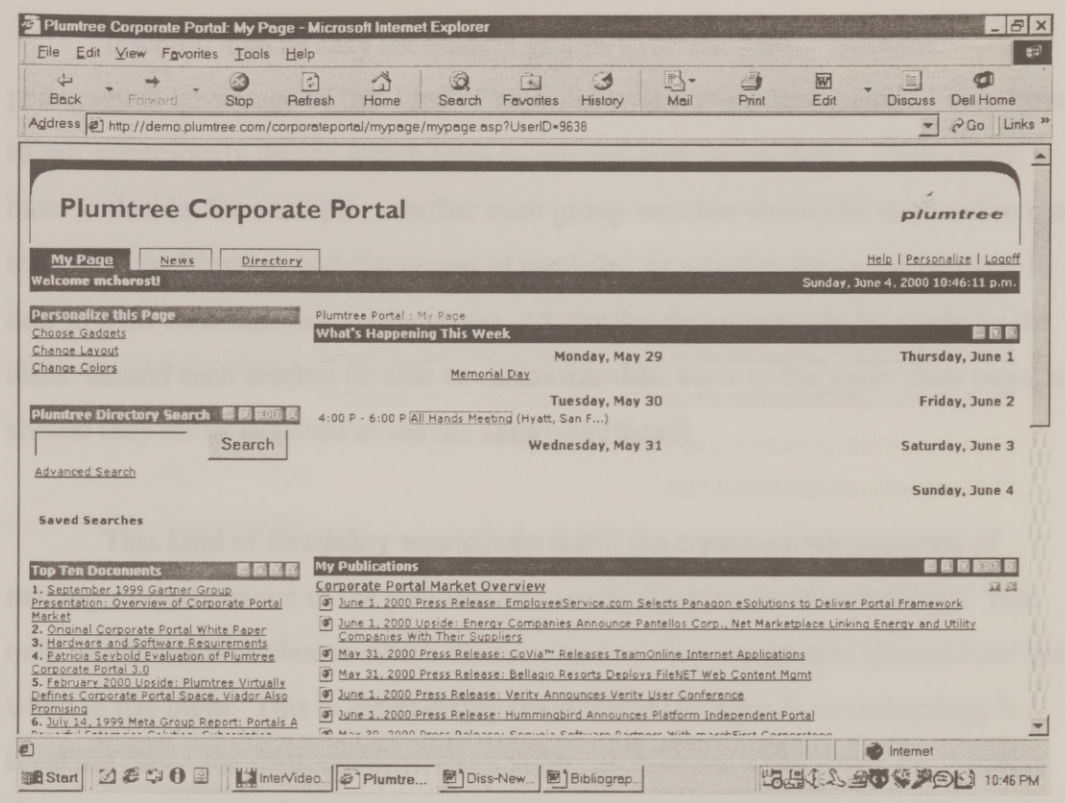


Figure 24: Plumtree's portal interface.

There would be an open protocol for writing new modules, so that designers could easily implement them without having to worry about integrating them with the environment. An open architecture like this would enable teachers to quickly assemble the tools that meet their needs. A modular platform along these lines would seem to allow teachers the right combination of customization, power, simplicity, and flexibility. Since assembling modules from scratch might be intimidating to new teachers, a school system might build a default "classroom" which teachers could then customize for their own courses.

It would also be easy for student groups to do the same. A major pedagogical advantage of this kind of modularity is that student groups would have to self-consciously decide which tools to use and how to use them. They might have to decide, for example, whether each group member should be required to use the same combination and placement of modules, or whether they could be free to design their own interface to the group. (A similar decision might be made by the class: should each student be able to customize their view of the main class page, or should they all be required to see the same interface?)

This kind of flexibility would help fulfill the constructivist program of making students aware of the political and social dimensions of technology. Not only might students choose which modules to use, they would also choose how and when to use them. This is increasingly important in an age where technology is breaking down the distinction between public and private time, given the intimate and universal availability implied by beepers, cell phones, and wireless PDAs. One group may choose not to exchange messages on weekends; another may establish a protocol of signifying availability and nonavailability; yet another may create a scale to indicate urgency and need for response. It would be hard to find a better way to help students learn how to manage their projects and balance competing demands.

A modular architecture would significantly facilitate autopoietic styles of teaching. This is because modules can be perceived as resources in an ecology of knowledge. Students and teachers could install and remove them at will, responding quickly as their needs change.

Extensibility

Hand in hand with modularity goes the concept of extensibility. The idea of modularity implies extensibility, because modules by definition are self-contained units which can be written relatively easily. Extensibility is important because it is a way that new knowledge about a social structure can be embodied in software.[11] There are two kinds of extensibility: knowledge extensibility and software extensibility.

Knowledge Extensibility

Marlene Scardamalia and Carl Bereiter (1996) have articulated a plausible design for software-facilitated "knowledge-building communities" (p. 254) in their description of CSILE, a groupware application designed to help students articulate ideas and build on the articulations of others. CSILE has a database at its core, to which students can add notes of various kinds (e.g., text or images). The software "serves as an objectification of a group's advancing knowledge…with additional facilities for reframing ideas and placing them in new contexts" (p. 261). Context is partially addressed by a "publication" rubric, where students are required to explain the utility of an insight or other piece of information and submit it to peer review before endorsement (263). The database continues to grow from semester to semester. "Thus, as in the real world," Scardamalia and Bereiter note, "each generation does not have to rediscover everything that the previous generation found out, but can instead attempt to go beyond it" (p. 264). When groups become

[11] Creighton (1998) has articulated a similar design objective for a physical space intended to support teamwork with configurable walls and furniture. "Virtually all walls -- except the outside walls of the building -- are movable. Interactive processes in particular require flexible spaces and the ability to adapt these spaces, in real time, as groups progress through different stages" (p. 99).

able to build their knowledge into the system so that they and future groups automatically benefit from it, the organization has embarked on a kind of intellectual capitalism (Neilson, 1997, p. 32).

A number of researchers have attempted to find socially meaningful forms of storing community knowledge, mostly in the form of stored and annotated narrative (Brown & Duguid, 2000; Kleiner & Roth, 1998). However, a major problem with any accumulative design is that the stored material quickly grows too overwhelming to navigate. After a certain size threshold is reached, it becomes almost impossible to know which items are valuable, which are outdated, which are trivial repetitions of earlier items, and which are simply wrong.[12] The essential problem is that it is easy to order computers to remember; it is harder to figure out what to tell them to forget.

Forgetting is not simply loss, as the information-processing perspective would view it; it is important to cultural memory. Mary Douglas (1986) argues that in culture, forgetting is part of an active process of re-evaluation and re-construction. She summarizes anthropological studies of Nuer genealogies, which observers noted developed gaps unrelated to reciters' inherent mental recall:

> Somewhere after the tribal founder and his two sons and his four grandsons and his eight great grandsons, the tribal memory has developed a yawning

---

[12] This was the single largest problem with the intranet at the eBusiness consultancy mentioned above. Their online repository of documents had worked well when the company had 200 employees, but it became disordered and difficult to use when the company had grown to eight times that size. One solution was to evaluate older items and relegate the less useful ones to an archive, but there were two problems with it. First, the work would have been extremely time-consuming. Even simply categorizing the documents proved to be a monumental task. Second and more fundamentally, what criteria were to be used in that the evaluation, and who would devise them? Part of the problem, also, was that the company rewarded contributions to the repository, but not deletions from it.

> hole, and multiple ancestors are tumbling headlong into it. They are not being forgotten randomly. The strengths and weaknesses of recall depends on a mnemonic system that is the whole social order (p. 72).

Douglas explains that the genealogies have important political consequences for weddings and inheritances – they are used to validate claims to family loyalty. There are certain specific rules for adjusting genealogies to demonstrate loyalty to a particular family, by which inconvenient sets of ancestors may be pruned or removed. By this selective forgetting, the social coherence of families and the society is preserved. Therefore, a genuinely useful system for archiving knowledge would also include tools for editing, pruning, consolidating, rewriting, and outright deleting material. In the case of an application like CSILE, it would be important to have some kind of software-facilitated pruning process. This could consist of an application to help students review a database at the beginning or end of a semester, systematically assessing its documents and archiving the ones deemed least valuable. Credit could be given not only for adding documents, but also for archiving, redacting, and consolidating documents.

Software Extensibility

In addition to extending bodies of knowledge, we also have the ability to extend the processes by which knowledge itself is created and used. Software extensibility allows software to evolve into increasingly rich representations of social activity. It also addresses a problem with modularity, e.g., that modules may not meet the needs of particular teachers and students (Repenning et. al., 1999, p. 478).

In a call for papers, Michael Koch (1998) writes, "...it is crucial to offer tailorability to the end users who perform the group processes supported by the software. 'The people performing the process must own the process'...since they know best about the process, its requirements, and individual and collective needs" (email, September 25, 1998). Similarly, theorists of groupware insist that any given program must undergo continuous development if it is to remain viable. Since work environments are always changing, the software must keep pace or become irrelevant. When talking about groupware design, Ciborra (1996) uses the word "care," by which he means "common sense, familiarity, and continuous commitment" (p. 6). It implies fine-grained and robust attention to the success of a particular initiative, a human quality of attention and nurturing. The ability to modify modules and create new ones is what keeps software relevant and alive. Extensible virtual spaces historically have enjoyed an extremely high level of user buy-in and commitment. MUDs and MOOs, Habitat, Quake – all of these environments have spawned enthusiastic user involvement, with users availing themselves of relatively simple tools to modify and extend the existing domains.

An important experiment in software extensibility is SRI's ESCOT, an effort aimed at creating modular and interoperable software tools to facilitate K-12 mathematics education. ESCOT's creators use the analogy of building a stereo system out of components, as opposed to actually building the components themselves (Roschelle, DiGiano, Pea, & Kaput, 1998). They are experimenting with components that can be combined to create activities and simulations for conveying mathematical concepts. From the example below, it can be seen that common design elements such as text boxes and graphs have been combined to form simulation applications.

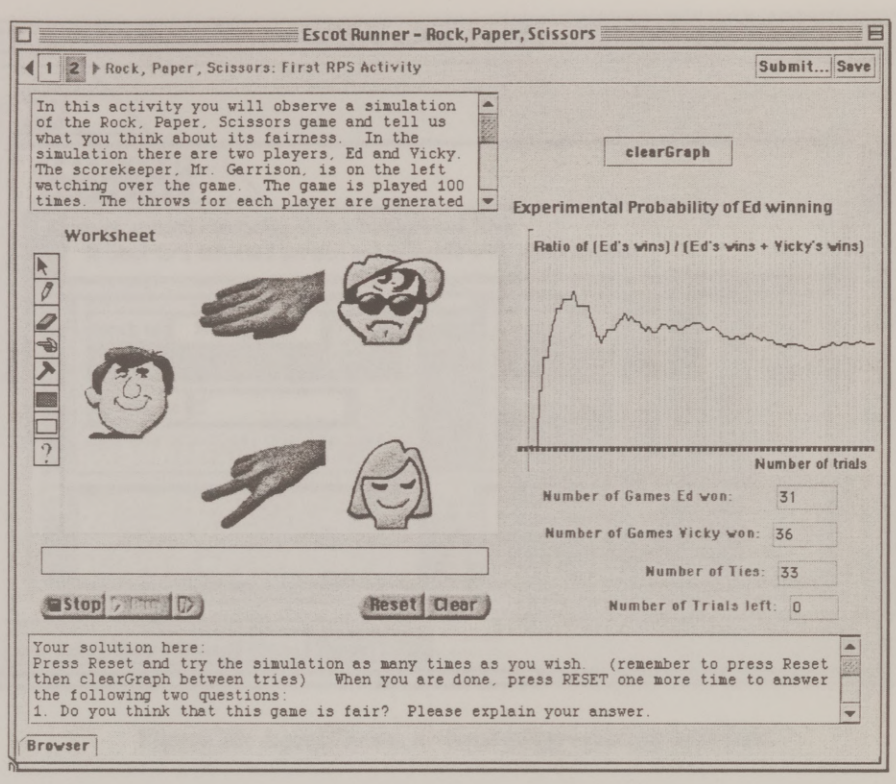


Figure 25: An ESCOT application, "Rock, Paper, Scissors."

The simulations are built in an authoring environment which allows users to visually assemble, observe, and modify the behavior of simulation elements (AgentSheets website, 2000). The authoring environment allows users to select objects and attach properties and rules to them. In the screenshot below, a simulation of an epidemic is being built. The object, a "person," is being programmed with rules which specify how it moves around on a grid and its probability of being infected by proximity to a sick person.

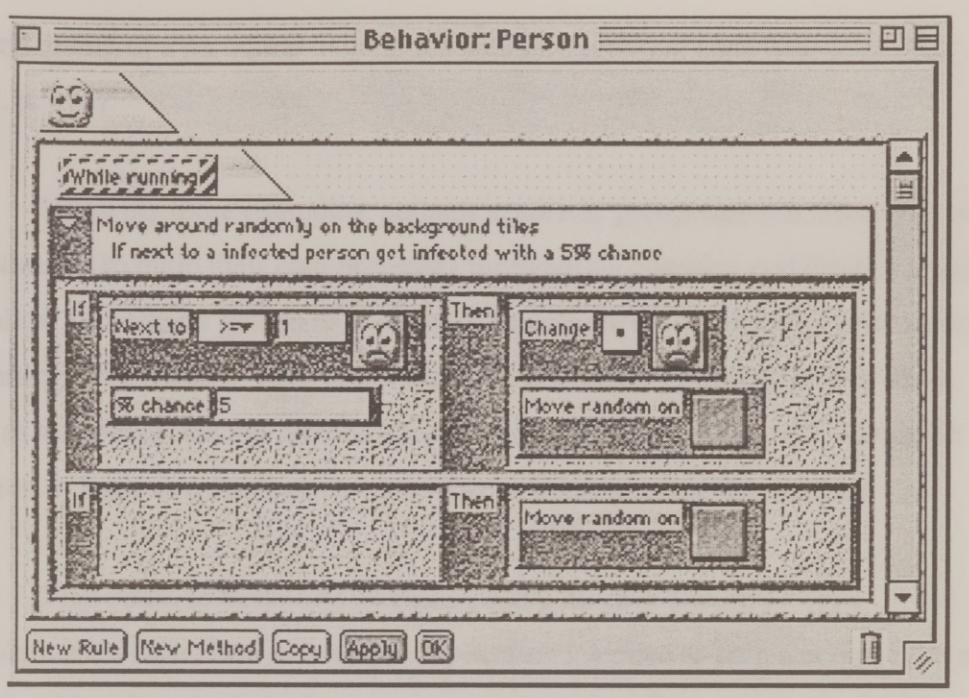


Figure 26: AgentSheets, a visual programming interface.

The AgentSheets technology was designed to facilitate the design of simulations of physical processes. Can similar toolsets be built for building applications to help students design and coordinate work processes? This idea has already been attempted in industry; Lotus Notes has a visual scripting language which allows users to build new routines for handling workflow. By selecting from an array of predefined components, such as drop-down boxes, text input boxes, and "send-this-to" commands, even relatively unsophisticated Notes users can set up routines which control the flow of documents and requests within a workgroup. NegotiationLens has similar capabilities. Users can create agents to automatically facilitate parts of a negotiation (e.g., automatically sending out proposals to a given group of people, with comment fields). In this sense, it allows users to build their

understanding of a particular negotiation task into software (Adelson & Jordan, 1992, p. 478).

Following these examples, perhaps academic groupware can allow teachers and students to build their own processes for handling complex collaborative tasks, and allow subsequent groups of students to modify them for their own ends instead of starting from scratch. Like AgentSheets, it could offer a library of primitives such as text boxes, submit buttons, drop-downs, and so on, along with automatic creation of back-end databases to store state information and user actions.

What could students and teachers build with such a library? As a speculative example, here is a simple application intended to help students hash out differing ideas for a paper's thesis statement. It could offer areas on the screen where students can register differing opinions and have them seen by the rest of the team, on an equal basis. In the screenshot below (which is a mockup), the first row of boxes offers each member of the team an individual space to propose a thesis. The page might conceal the other team members' contributions until all the members have weighed in, in order to require each member to think independently. The second row consists of a single box, where team members can play with text freely, collectively building a thesis which incorporates the best aspects of each contribution.

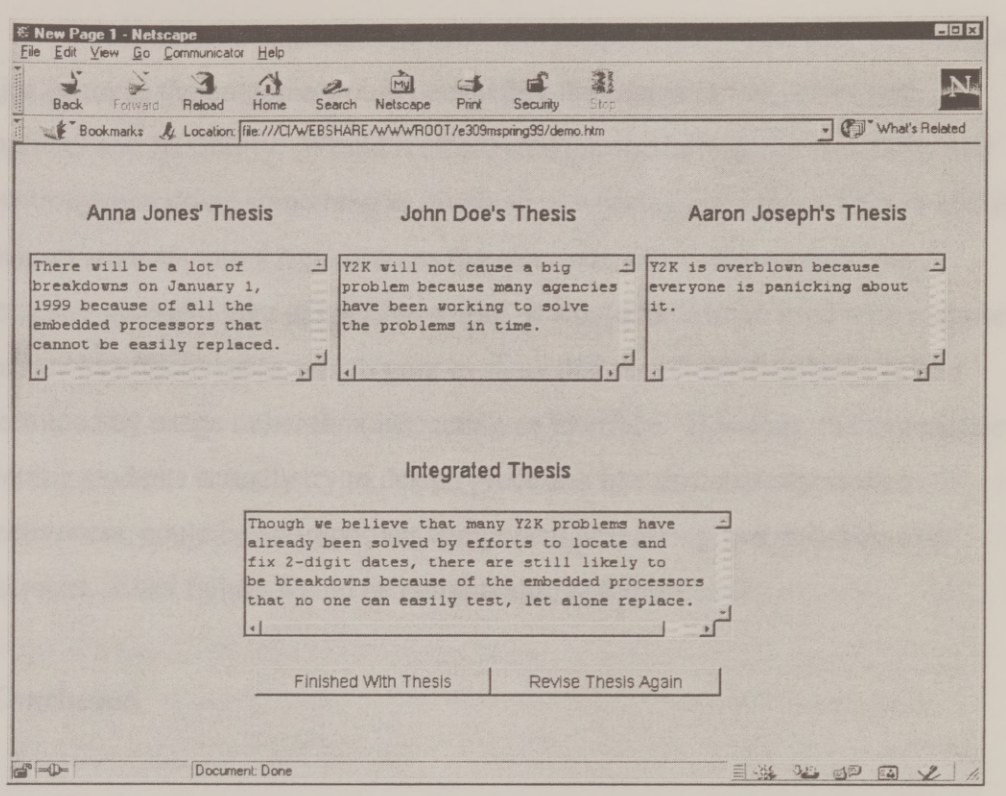


Figure 27: An interface for collective thesis creation.

It may be that the team will be satisfied after this first iteration. However, if any member of the team thinks there is room for improvement, he or she can call for a second iteration, by clicking on the "Revise Thesis Again" button. This will generate another set of boxes, calling on team members to enter responses to the group thesis and then to build a second collective thesis. This process can be repeated as often as necessary. As with The Coordinator and NegotiationLens, the design does not constrain *what* students enter. It simply gives them a space where they can enter proposed theses and subsequently refine them, iterating as often as necessary.

Would such an application work? Only experience would tell. Probably a major factor in the outcome would be whether the design in fact offers both simplicity and flexibility. Would it afford students the latitude for flexibility and invention when doing something as involved as working out a thesis? Or would it shoehorn students into a rigid process that does not effectively scaffold the interactive give-and-take of creative work? It would have to be tried with real users to assess its effectiveness, which goes to show that simplicity/flexibility is best determined by usage rather than inspecting an interface. However, the experience of letting students actually try to design processes like this, and assess their effectiveness, could be an important component of learning in a collaborative classroom. Even failure would be enlightening and instructive.

7. Conclusion

The four design principles outlined in this chapter, simplicity/flexibility, spatiality, modularity, and extensibility, were specifically chosen to aid designers in creating software which is able to function robustly in information ecologies. As I noted earlier, they strongly resemble the terms that Chapter 1 used to characterize information ecologies; it could be argued, in fact, that they are a rewriting of Chapter 1's descriptive terms into prescriptive ones. For example, local units are relatively simple, and their flexibility comes from the fact that they can be useful in many different contexts. (An email message can hold an infinite variety of content; a team room can support many different kinds of social activity.) This likeness is hardly surprising, given the often-noted isomorphisms which exist between ecological systems and computer networks. The best way to support an information ecology is to introduce tools which naturally fit into ecologies.

There is no route to success solely through software design, however. In this chapter I have argued that in order to function effectively in constructivist classrooms, software design has to be complemented by community systems design, which treats software as but one element in a complex ecology of actors and resources. In the end, the most influential designer in the classroom environment is not the programmer, but the teacher/manager.

Focusing on community systems design carries significant implications for teaching practice. Whereas in an earlier era teachers focused on *operating* software in order to transmit information, they now must focus *improvising with* software to design learning environments. The difference is profound; it is analogous to the difference between an assembler following instructions and a craftsman working with materials. Tools which are simple to grasp and applicable in multiple circumstances are better affordances of craft, and for that reason are likelier to be adopted than complex, specialized ones. They allow both teachers and students more opportunity for autonomy and collaborative activity in the classroom, and for that reason are better supports for the constructivist enterprise.

Appendix 1: Answers to Surveys on Activity Tracking

These are students' answers to the question on activity tracking posed in the end-of-semester surveys. The survey was conducted by having students download a Word file and save it to a network drive under a nonsense name (this ensured anonymity.) For each year, I have grouped the answers according to whether I believed they constituted a "strong yes," "weak yes," "weak no," or "strong no" answer to the question. The number in parentheses refers to the number of answers grouped in that category.

Survey Question in Fall 1997 and Spring 1998: "Did your knowledge of the monitoring tools (i.e. the recording of your logins and activity) change how often you used the site, and/or what you did on it?"

Fall 1997

Strong No (7)

- I was not affected by the the monitoring. I don't feel the number of logins should be used to evaluate the grade. I logged into the website only when I needed to access information such as class announcements, course assignments, etc.
- No
- No. I know that the data won't be used in any way to evaluate individual performance, just to see what parts of the site were most useful.
- No
- No

- No. I knew statistics were being collected on activity on the site. It didn't change my activity.
- For me no, for others yes. I didn't change my behavior knowing that I was being monitored because I felt it wouldn't have a positive effect on me or the instructor.

Weak No (2)

- No not really.
- Not really. I did think about using it more but then when you think (more) about it, it becomes to be more of a hassle that it's worth. The number can't be judged the same for everyone.

Weak Yes (5)

- Not really, if anything, I checked it less after I found that that it was being monitored but that no significant data was being taken from it.
- Yes, I logged on more at the very beginning and then I realized that it was silly.
- At first it did when I found out, then after the class discussion about it, I slowly went back to using it just when I needed to.
- The monitoring did affect my activity at first. The number of logins increased.
- Yes, a little only. I thought it might be seen as a participation grade.

Strong Yes (2)

- Yes, I logged in a couple of times just to up my stats.

- Yes I discontinued use of the website after I found out about it. I think you should announce that you would be anonymously tracking.

Other (1)

- Yes, I logged in more frequently when I found that that we were being monitored but there were many factors that affected this situation. That is about the time that I got my own computer and was able to have easier access and more frequent access to a computer. It was a lot easier to log on and work on a computer because I didn't have to coordinate with a friend's schedules.

Spring 1998

Strong No (7)

- No, it really had no effect.
- No, not all.
- No. I check all class sites/newsgroup (if the class has one) very frequently.
- No.
- No. I don't care who looks at me on the Web.
- No, I didn't realize that the data was recorded until long after the semester had begun. It didn't affect my use of the site at all.
- No, I don't think I ever looked at the logins and activity. I felt that I was logging in only when I needed to use the site. I didn't feel any pressure.

Weak No (4)

- Not really.
- No, I forgot they existed as the class progressed. I was apprehensive about them at first, but in time I didn't notice. They didn't change how or when or why I used the website.
- Not really but I did feel like I was being monitored and it made me feel a little apprehensive about the times I hadn't logged on in a while.
- Not really.

Weak Yes (3)

- At first it did, but towards the end I just went to the class website when needed.
- Yes, at first it did because I thought you were taking it into consideration when grading, but then I found that you didn't so I didn't think about it anymore.
- A little maybe. Sometimes I just needed to get some info so I would log in quickly to get both accomplished.

Strong Yes (2)

- After I found out about the recording of the logins etc., I felt compelled to log in more and click on everything in the site because I didn't want to fall behind compared to the class.
- Yes, I was more conscious of logging in. I felt like I had to or Mike would know I was not using the technology.

Other (2)

- (No answer)

- (No answer)

Survey Question in Fall 1998 and Spring 1999: "As you know, the site records your logins and many of your actions. Did this affect how often you used the site, and/or your actions?"

Fall 1998

Strong No (14)

- No not at all, if I thought the records would affect my grade I would have but I really don't see how they could.
- No, I didn't really care that my actions were logged
- No, not all.
- No, I usually wanted to check either the syllabus or group notes, so I logged in enough anyway.
- Not at all.
- No, record of my log in I don't think matters as long as I get my goal across.
- No.
- No.
- No.
- No.
- No, not really, I log in when I needed to. I did not log in just to show that I was logging in
- No, this didn't have an effect on my actions.
- No.
- No.

Weak No (1)

- It did not affect the way I used it, but I could see how someone would be more willing to log in every day just to get it shown that they were at the site. Since you can't really see what the person did at the site, I don't think it is a good way to judge whether somebody kept up with the class or not.

Weak Yes (0)

Strong Yes (1)

- Yes I tried to log in more times in order to appear to be visiting more than I was.

Other (0)

Spring 1999

Strong No (16)

- Not at all.
- NO.
- Didn't affect me at all.
- No it doesn't. I use it because I need to use it.
- No.
- No. I used the site whenever I need it.
- NO.
- Nope, not one bit.
- No not at all. It was interesting to see when people would log in though.
- No, it didn't matter to me.

- No. During collaborative writing, I would want to know if anyone in my group had logged on, so that I would have a cue to check for revisions in our work.
- No, not at all. I never felt that I was being watched or anything like that. I simply did what I needed to do. If it updated the latest events [module], that was fine. It really didn't matter to me.
- No.
- No, this did not affect my actions at all. I often forgot that it was there, and when I did see it, I would rarely pay any attention to it.
- Not at all. I really could've cared less.
- No, it did not affect me in any way.

Weak No (3)

- No. It did not concern me too much.
- Not really. I didn't think the instructor would check how often a certain student checked into the Collaboration Center.
- Not really. Since I really don't care about such a trivial thing.

Weak Yes (1)

- At first, I was checking up constantly just to make sure that this wasn't used against my grade - a way to try to make sure I got recognition for my interest in the course. Then, as the class progressed, I'd check in only as I needed to. As our groups settled in and I began to know some of my classmates, ICQ became a more common form of interaction with the people that I was working with, instead of the Collaboration Center.

Strong Yes (1)

- Yes. When I discovered how many times I logged in at the CC home (pretty high), I began to keep the window open to the CC on my computer for most of the time so that I could keep the number low. For some reason I felt that logging in too many times would higher expectations from the teacher (?). I don't know, truthfully... but it did lessen the times I refreshed or logged in.

Other (1)

- I believe so... it is really fun to log in this site. And it is a really efficient way to communicate with students. But the only thing I worried is posting information does not guarantee to let them have that information. Because of this I missed some assignments.

Appendix 2: The Syllabus of English 309, Fall 1999

This is taken from the Syllabus module of the Collaboration Center.

Week 1	Tue, Jan. 19	Color code: Red: Major assignments Green: Knowledge Workshops Blue: Major due dates Introduction to course **Assignment:** Add yourself to the Roster by going to the Web site (http://corax.cwrl.utexas.edu/cc), going to "New Students," and logging yourself in. **Assignment:** Fill out the Scavenger Hunt worksheet to familiarize yourself with the Collaboration Center. Due Thursday. **Assignment:** Take the survey titled "Questions for Entering Students." Due before Thursday.
	Thu, Jan. 21	**QuickCam pictures** Release form paperwork Assignment: Project 1a: "Design Your Ideal Work Environment" Groups formed for Project 1a. **Reading** for Tuesday: Wanda Orlikowski, "Evolving with Notes: Organizational Change around Groupware Technology" (handout.) Read this before you start work on Project 1a. Before Tuesday, make one thoughtful comment about Orlikowski's article in the "Orlikowski Discussion" thread in the Forums. Try to include responses to other students in your comment. Length: 1 paragraph.
Week 2	Tue, Jan. 26	**Class Activity:** Presentation of Project 1a Project 1b assigned, "Create A Survey" (survey due Thursday; analysis due next Tuesday)
		Discussion (f2f): Wanda Orlikowski's article. We'll base our discussion on the things people said in the Forums. **Due:** Surveys should be published so that the class can take them **Assignment:** Take the surveys developed by each project

		group **Assignment:** Each group should discuss its survey's results and prepare a report on them, for Tuesday. Place the report in your group's Notepad. Each person in the group should be involved in writing the report. Length: 2-3 paragraphs. **Reading for Tuesday:** Kitty O. Locker, "What Makes A Collaborative Writing Team Successful?" **and** Janis Forman, "Literacy, Collaboration, and Technology" (handouts.) Before Tuesday, make one thoughtful comment about each article in the relevant Forums thread. Try to include responses to other students in your comments. Length: 1 paragraph each.
Week 3	Tue, Feb. 2	**Due:** Writeups of Surveys (circulate to the class via email; we won't do presentations on them.) **Discussion** (f2f): "What Makes Good Collaboration Happen?" We'll cover both the Locker and Forman readings. We'll base our discussion on the things people say in the Forums. Knowledge Workshop: Word 97's Collaborative Tools **Assignment:** Read the following two Knowledge Base articles: "Reflections on Good Collaboration" (Spring 1999) "Collected Wisdom on Collaborative Writing" (Fall 1998) **Homework:** Finish the KW out of class if you don't have a chance to finish it in class.
	Thu, Feb. 4	**Short Meetings** (15 minutes) with Project 1c groups Guest Speaker: Jami Castillo and Rose Greer, IBM Information Design and Development Project 1c: Reflective Essays (using Word 97's collaborative tools.) See assignment in Knowledge Base. Due next Tuesday. **Reading:** Chapter 4, "Four Skills of Reflective Listening," from Robert Bolton's *People Skills* (handout), to prepare you for the "Good Communication Skills" workshop next class.
Week 4	Tue, Feb. 9	Group Work Time: Project 1c (40 minutes; emphasize reflective listening.)
	Thu, Feb. 11	**Due:** Project 1c Project 2 Assigned: The Magazine Paper or Knowledge

		class's needs in the topic you propose to teach.
Week 5	Tue, Feb. 16	**Assigned Reading:** Chapter 2 of Mike Chorost's dissertation Discuss Forums and Surveys, coordinate project groups
	Thu, Feb. 18	**Magazine Paper:** Meet in groups to discuss magazine articles. In the assignment I haven't asked you to have a full complement of articles ready, but it sure as heck will help if you do. See if you can have a working thesis by the end of the meeting; put it in the group's Notepad. Your next goal is to have a first draft ready by Feb 25th. Final draft due Mar 4th. **Knowledge Workshop:** Analyze your survey results and begin designing the workshop. Have an outline of the KW by the end of the meeting; put it in your group's Notepad. Your next goal is to have a first draft of the handout ready by Feb 18th. KW teams: Arrange a rehearsal date with me.
Week 6	Tue, Feb. 23	**Discussion:** Chapter 2 of Mike's dissertation (in chat room) Group Work Time If there's time: Effective organization, transitions between sections/paragraphs.
	Thu, Feb. 25	**Due:** First Draft of Magazine Paper / KW. **Discussion**: Brief discussion of Virtual Class transcripts Schedule KWs Bring enough copies for each member of your team, and me. (You can turn it in at the end of class – you'll have time to work on the details in class.) In the margins, your team should write notes showing which person did what. In other words, next to each paragraph or section should be the names of the person or persons who worked on it. Instructor returns comments on drafts by Sunday. Schedule midterm conferences (click on the button for "Special" conferences to see the available timeslots.)
Week 7		

	Thu, Mar. 4	Project 2 Due **Discussion:** How to Get WNT accounts; coordinating Web site work **Project 3 Assigned:** Web Debate (first draft due 3/23) Set up project groups and threads in Forums by Sunday Knowledge Workshop 1: Creating a Basic Web Site (Example of finished site) Knowledge Workshop: Internet Research (Chorost)
Week 8	Tue, Mar. 9	Discussion: Organizing Web sites, what a "first draft" is, tips on writing for the Web In-class Survey: Which KW will you attend? Group Work Time: Convert your group's thread into an hypertextual map of your site. **Reading:** Two articles by Jakob Nielsen: "How Users Read on the Web" "Be Succinct!" I'll probably put up a quiz so you can review your understanding of these articles.
	Thu, Mar. 11	Due: Midterm Analysis and Evaluation First Student-Run Knowledge Workshop: Computer Parts (Sang Ham and Sheney Chen) and Networking (Steven Faust and Jeremy Bopp) **Assignment:** Take post-workshop survey or quiz
Week 9	Tue, Mar. 16	SPRING BREAK – NO CLASS
	Thu, Mar. 18	SPRING BREAK – NO CLASS
Week 10	Tue, Mar. 23	45 minutes: Second Student-Run Knowledge Workshop: Effective Web Design (Alan Phan and Jackson Nghiem) 30 minutes: Group Work Time on web sites **Assignment:** Take post-workshop surveys
	Thu, Mar. 25	20 min: Discussion – "What's Good and Bad about Computers?" 55 min: Group Work Time (focus on site map, interface, technical issues of creating and uploading skeleton site)
Week 11	Tue, Mar. 30	First draft of Project 3 due (final due Apr. 6th)

		5-7: Rehearsal of Flash KW
	Thu, Apr. 1	TBA
Week 12	Tue, Apr. 6	Group Work Time
	Thu, Apr. 8	Final draft of Project 3 due Third Student-Run Knowledge Workshop: Flash (Henry Tang and Roman Baltazar) **Assignment:** Take post-workshop survey or quiz **Project 4 Assigned:** Individual Web Project (now updated for Spring 1999.) **Assignment:** For Tuesday, prepare a prospectus for your Web site.
Week 13	Tue, Apr. 13	Discussion: Previous Web projects; criteria of assessment. **Due:** Prospectus for Project 4 Reading for Thursday's Virtual Class: Neil Postman, "Informing Ourselves to Death"
	Thu, Apr. 15	Virtual Class (go to the CC's chat room from wherever you are!) Don't forget to file your taxes, or the Man will jail you and you'll have difficulty completing your Web project. Instructor returns comments on Project 4 prospectuses
Week 14	Tue, Apr. 20	Work time; clarification of assignment; discussion of projects with individual students
	Thu, Apr. 22	**Optional class:** come if you need help, want to get work done, or simply get the warm fuzzies from being in class.
Week 15	Tue, Apr. 27	First draft of Project 4 due Fourth KW: Netscape (Kareem Badr and Vazir Sultan) I'll also present short tutorials on Photoshop on both Tuesday and Thursday so that all interested parties have a chance to learn the basics of scanning photographs.
	Thu, Apr. 29	Instructor schedules final conferences. Assignment: Final Self-Evaluation The rest of the class I'll spend time checking out each person's site to make sure they're making progress.
Week	Tue, May. 4	Project 4 due

16		I will be in FAC 9a from 12:30 until class time tomorrow, to help anyone who needs it. The plan for today is mostly work time, for you to work on the last tough bits of your sites and get help resolving them. For those who are having tough problems, I'll invite them to show the problem to the class and ask if anyone knows how to solve it. Once your site is finished, please post the URLs of your Web Debate and Final Project in the Long Roster.
	Thu, May. 6	Final Self-Evaluations due Survey: Evaluation of course Party!!

Bibliography

ActionTechnologies Web site (2000). Retrieved December 1, 2000 from http://www.actiontechnologies.com/Corporate/Background/index.cfm.

Adams, L., Toomey, L., & Churchill, E. (1999, January). Distributed research teams: meeting asynchronously in virtual space. In *Proceedings of the Thirty-second Annual Hawaii International Conference on System Sciences* (Wailea, Hawaii). Retrieved July 2nd, 1999 from http://www.fxpal.xerox.com/PapersAndAbstracts/papers/ada99.pdf.

Adelson, B. & Jordan, T. (1992). The need for negotiation in cooperative work. In E. Barrett (ed.), *Sociomedia: Multimedia, Hypermedia, and the Social Construction of Knowledge*, pp. 469-492. Cambridge, MA: MIT Press.

AgentSheets Web site (2000). AgentSheets backgrounder. Retrieved December 1, 2000 from http://www.actiontechnologies.com/Corporate/Background/index.cfm.

Agre, P. (1996, March). The Internet and education. Retrieved September 10, 1999 from http://communication.ucsd.edu/pagre.

Agre, P. (1998). Beyond the mirror world. In P. Agre & M. Rotenberg (eds.), *Technology and privacy: the new landscape*. Cambridge, Mass: MIT Press.

Alexander, B., Crowley, J., Lundin, D., Mudry, V., Palmer, S., & Rabkin, E. (1997, September). E-COMP: a few words about teaching writing with computers. *THE Journal*, pp. 66-67.

Allen, M. (1999). *Assessing Effectiveness in Corporate Universities.* Unpublished doctoral dissertation, University of Southern California.

American Association of University Women Educational Foundation (2000). Tech-savvy: educating girls in the new computer age. Retrieved June 7th, 2000 from http://www.aauw.org/2000/techsavvy.html.

Attewell, P. & Rule, J. (1998). Computing and organizations: what we know and what we don't know. In I. Greif (ed.), *Computer-Supported Cooperative Work: A Book of Readings*. San Mateo, CA: Morgan Kaufmann Publishers, Inc.

Bailey, J. (1996). *After Thought: The Computer Challenge to Human Intelligence.* New York: Basic Books.

Bair, J. (1997). The role for technology in knowledge management. In S. C. Hayne & W. Prinz (eds.), *Proceedings of GROUP '97*, pp. 3-6. New York: Association for Computing Machinery.

Balla, J. & Andrews, L. (1999, September). Backweb sales accelerator brings KM to the sales chain. *KM World 8:9*, p. 32.

Bardram, Jakob E. (1997). I love the system – I just don't use it! In S. C. Hayne & W. Prinz (eds.), *Proceedings of GROUP '97* (pp. 251-260). New York: Association for Computing Machinery.

Barker, T. & Kemp, F. (1990). Network theory: a post-modern pedagogy for the writing classroom. In C. Handa (ed.), *Computers in Society: Teaching Composition in the Twenty-first Century*, pp. 1-28. Upper Montclair, NJ: Boynton-Cook.

Barr, M. & Syverson, M. A. (1995). What is the online learning record? Retrieved December 6, 2000 from http://www.cwrl.utexas.edu/~syverson/olr/intro.html.

Bass, R. (1999, January 22). Discipline and publish: faculty work, technology, and accountability. Plenary address, AAHE Forum on Faculty Roles and Rewards, San Diego. Available http://www.georgetown.edu/bassr/disc&pub.html.

Bell, P. (1997). Using argument representations to make thinking visible for individuals and groups. In R. Hall, N. Miyake, & N. Enyedy (eds.), *Proceedings of CSCL '97: The Second International Conference on Computer Support for Collaborative Learning*, pp. 10-19. Toronto: University of Toronto Press. Retrieved February 7, 1998 from http://www.oise.utoronto.ca/cscl/papers/bell.pdf.

Bellotti, V. & Bly, S. (1996). Walking away from the desktop computer: distributed collaboration and mobility in a product design team. In *Proceedings of CSCW '96*, pp. 209 – 218. New York: Association for Computing Machinery.

Benedikt, M. (1991). Cyberspace: some proposals. In M. Benedikt (ed.), *Cyberspace: First Steps*, pp. 119-224. Cambridge, MA.: MIT Press, 1991.

Bennett, F. (1999). *Computers as tutors: solving the crisis in education.* Faben, Incorporated. Retrieved May 6, 1999 from http://www.cris.com/~faben1/section5.shtml.

Birkerts, S. (1994). *The Gutenberg Elegies: the fate of reading in an electronic age.* Boston: Faber and Faber.

Bock, G. E. & Marca, D.A. (1995). *Designing groupware: a guidebook for designers, implementors, and users.* New York: McGraw-Hill.

Bolter, J. D. (1991). *Writing space: the computer, hypertext, and the history of writing*. Mahwah, NJ: Lawrence Erlbaum Associates.

Bolton, R. (1979). *People skills: how to assert yourself, listen to others, and resolve conflicts.* New York: Touchstone (Simon & Schuster).

Bonk, C. J. & Cunningham, D. J. (1998). Searching for learner-centered, constructivist, and sociocultural components of collaborative educational learning tools. In C. J. Bonk & K. S. King (eds.), *Electronic collaborators: learner-centered technologies for literacy, apprenticeship, and discourse*, pp. 25-50. Mahwah, NJ: Lawrence Erlbaum Associates.

Bourdieu, P. (1977). *Outline of a theory of practice*. Cambridge: Cambridge University Press.

Bronner, E. (1999, July 11). Left and right are crossing paths. *New York Times*. Retrieved July 11, 1999 from http://www.nytimes.com/library/review/071199partisanship-review.html.

Brown, J. S. & Duguid, P. (1994). Borderline issues: social and material aspects of design. *Human-Computer Interaction, 9*, pp. 3-36.

Brown, J. S. & Duguid, P. (1995). The social life of documents. Retrieved July 20th, 1999 from http://www.parc.xerox.com/ops/members/brown/papers/sociallife.html.

Brown, J. S. & Duguid, P. (1995). Universities in the digital age. Retrieved July 20th, 1999 from http://www.parc.xerox.com/ops/members/brown/papers/university.html.

Brown, J. S. & Duguid, P. (2000). *The social life of information.* Cambridge, MA: Harvard Business Press.

Brown, J. K. (1999, February). Technology lends mentoring programs a hand. *Converge Magazine 2:2*, pp. 42-43.

Bruffee, K. (1982). Liberal education and the social justification of belief. *Liberal Education, 68,* pp. 95-114.

Bruffee, K. (1984). Collaborative learning and the 'conversation of mankind.' *College Education, 46,* pp. 635-652.

Bruffee, K. (1993). *Collaborative learning.* Baltimore, MD: Johns Hopkins University Press.

Bullen, C. V. & Bennett, J. L. (1990). Learning from user experience with groupware. In *Proceedings of CSCW '90*, pp. 291-302. New York: Association for Computing Machinery.

Cameron, T., Barrows, H., & Crooks, S. (1999). Distributed problem-based learning at Southern Illinois University School of Medicine. In C. Hoadley (ed.), *Proceedings of CSCL '99*, pp. 86-93. New York, NY: Association for Computing Machinery.

Castells, M. (1996). *The rise of the network society.* Oxford: Blackwell Publishers Ltd.

Ciborra, C. (1996). What does groupware mean for the institutions hosting it? In C. Ciborra (ed.), *Groupware and teamwork: invisible aid or technical hindrance?*, pp. 1-19. West Sussex, England: John Wiley & Sons.

Cole, P. & Johnson, E. (1996). Lotus development: Team Room - a collaborative workspace for cross-functional teams. In P. Lloyd & R. Whitehead (eds.), *Transforming organizations through groupware: Lotus Notes in action*, pp. 23-38. Berlin, Heidelberg, New York: Springer-Verlag.

Cox, B. (1998). Evolving a distributed learning community. Retrieved January 27th, 2000 from http://www.virtualschool.edu/cox/OnlineClassroom.html.

Creighton, J. (1998, September/October). Meeting rooms of the future. *Group Computing*, pp. 28-31, 60.

Davenport, T. (1997). *Information ecology: mastering the information and knowledge environment.* Oxford: Oxford University Press.

Davis, J. R. & Huttenlocher, D. P. (1995). Shared annotation for cooperative learning. In *Proceedings of CSCL '95.* Retrieved June 9th, 1998 from http://www-cscl95.indiana.edu/cscl95/davis.html.

De Graff, R., Gerrits, H., & Huysman, M. (1999, December 14). Mutual impact of groupware application and learning and communication processes. Poster session, CSCL '99.

DeLong, S. E. (1997, March). The shroud of lecturing. *First Monday, 2:5.* Retrieved November 12, 1998 from http://firstMonday.dk/issues/issue2_5/DeLong.

Dewey, J. (1916, 1997). *Democracy and education.* New York: Simon and Schuster.

Dibbell, J. (1996). A rape in cyberspace: or, how an evil clown, a hatian trickster spirit, two wizards, and a cast of dozens turned a database into a society. In V. Vitanza (ed.), *CyberReader*, pp. 448-465. New York: Allyn and Bacon.

Douglas, M. (1986). *How institutions think.* Syracuse, New York: Syracuse University Press.

Drucker, P. (1998). The coming of the new organization. In *Harvard Business Review on Knowledge Management*, pp. 1-20. Boston: Harvard Business School Publishing.

Duffy, T. M., Dueber, B., & Hawley, C. Critical thinking in a distributed environment: a pedagogical base for the design of conferencing systems. In C. J. Bonk & K. S. King (eds.), *Electronic Collaborators: Learner-Centered Technologies for Literacy, Apprenticeship, and Discourse*, pp. 51-78. Mahwah, NJ: Lawrence Erlbaum Associates.

Ede, L. & Lunsford, A. (1985). Let them write – together. *English Quarterly*, *18*, pp. 119-27.

Elbow, P. (1973). *Writing without teachers.* New York: Oxford University Press.

Engelbart, D. (1988). Authorship provisions in AUGMENT. In I. Greif (ed.), *Computer-supported cooperative work: a book of readings*. San Mateo, CA: Morgan Kaufmann Publishers, Inc.

Engeström, Y & Escalante, V. (1996). Mundane tool or object of affection? The rise and fall of the Postal Buddy. In B. A. Nardi (ed.), *Context and consciousness: activity theory and human-computer interaction*, pp. 325-373. Cambridge, Mass: MIT Press.

Faigley, L. (1992). *Fragments of rationality: postmodernity and the subject of composition*. Pittsburgh, PA: University of Pittsburgh Press.

Feenberg, A. (1999, February 9). Distance learning: promise or threat? Red Rock Eater News Service. Retrieved May 1, 2000 from http://www.tao.ca/wind/rre/0622.html.

Fellman, M. W. (1999, Fall). Breaking tradition. *Marketing Research*, pp. 21-24.

Feltovich, P. J., Spiro, R. J., Coulson, R. L., & Feltovich, J. (1996). Collaboration within and among minds: mastering complexity, individually and in groups. In T. Koschmann (ed.), *CSCL: theory and practice of an emerging paradigm*, pp. 25-44. Mahwah, NJ: Lawrence Erlbaum Associates.

Fisher, K.,& Fisher, M. (1997). *The distributed mind: achieving high performance through the collective intelligence of knowledge work teams.* New York: AMACOM.

Fjuk, A. (1995). Towards an analytical framework for CSCdistanceL. In *Proceedings of CSCL '95*. Retrieved April 17, 1998 from http://www-cscl95.indiana.edu/cscl95/fjuk.html.

Flower, L. (1993). *Problem solving strategies for writing*, 4th ed. Fort Worth: Harcourt Brace Jovanovich.

Forman, J. (1994). Literacy, collaboration, and technology: new connections and challenges. In C. Selfe & S. Hilligoss (eds.), *Literacy and computers: the complications of teaching and learning with technology*, pp. 130-143. New York: Modern Language Association.

Foucault, M. (1977). *Discipline and punish: the birth of the prison.* New York: Pantheon Books.

Friedman, T. (1993). Making sense of software. Retrieved November 30th, 2000 from http://www.virtualschool.edu/mon/SocialConstruction/MakingSenseOfSoftware.

Fuhrer, U. (1996). Behavior setting analysis of situated learning: the case of newcomers. In S. Chaiklin & J. Lave (eds.), *Understanding Practice: Perspectives on Activity and Context*, pp. 179-214. Cambridge University Press.

Galegher, J., & Kraut, R. E. (1990). Computer-mediated communication for intellectual teamwork: a field experiment in group writing. In *Proceedings of CSCW '90*, pp.65-78. New York: Association for Computing Machinery.

Garvin, D.A. (1998). Building a learning organization. In *Harvard Business Review on Knowledge* Management, pp. 47-80. Boston: Harvard Business School Publishing.

Bennahum, D. (1996, November 11). David Gerlenter: The InterMinds interview. ElectricMinds. Retrieved December 6, 2000 from http://www.abbedon.com/electricminds/html/tom_interminds_1108.html.

Gill, Z. (1999). A design method using Darwinian principles for collaborative problem-solving environments (CPSES). In C. Hoadley (ed.), *Proceedings*

of CSCL '99, pp. 203-210. New York, NY: Association for Computing Machinery.

Gleick, J. (1987). *Chaos: the making of a new science.* London: Sphere Books.

Goodman, G. O. & Abel, M. J. (1987). Communication and collaboration: facilitating cooperative work through communication. *Office: Technology and People*, *3*, pp. 129-145.

Graff, G. (1987). *Professing literature: an institutional history.* Chicago, IL: University of Chicago Press.

Graves, G., & Haller, C. (1994). The effect of secondary school structures and traditions on computer-supported literacy. In C. Selfe & S. Hilligoss (eds.), *Literacy and computers: the complications of teaching and learning with technology*, pp. 144-156. New York, NY: Modern Language Association.

Grinter, R. (1997). From workplace to development: what have we learned so far and where do we go? In S. Hayne & W. Prinz (eds.), *Proceedings of GROUP '97,* pp. 231-240. New York, NY: Association for Computing Machinery.

Guzdial, M. (1997). Information ecology of collaborations in educational settings: influence of tools. In *Proceedings of CSCL'97,* pp. 83-91. New York, NY: Association for Computing Machinery.

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. In C. Hoadley (ed.), *Proceedings of CSCL '99,* pp. 211-217. New York, NY: Association for Computing Machinery.

Hairston, M (1982). The winds of change: Thomas Kuhn and the revolution in the teaching of writing. *College Composition and Communication*, *33.1*, pp. 76-88.

Hara, N. & Kling, R. (1999, July 9). Students' frustrations with a web-based distance education course. Red Rock Eater web site. Retrieved March 12th, 2000 from http://www.tao.ca/wind/rre/0733.html.

Harasim, L. (1996). Online education. In T. M. Harrison & T. Stephen (eds.), *Computer Networking and Scholarly Communication in the Twenty-First Century University,* pp. 203-214. Albany, NY: SUNY Press.

Haraway, D. (1991). A cyborg manifesto: science, technology, and socialist-feminism in the late twentieth century. In *Simians, Cyborgs and Women: The Reinvention of Nature,* pp. 49-181. New York, NY: Routledge Kegan & Paul.

Hargrove, R. (1998). *Mastering the art of creative collaboration*. New York, NY: McGraw-Hill.

Hartman, K., Neuwirth, C., Kiesler, S., Sproull, L., Cochran, C., Palmquist, M., & Zubrow, D. (1991). Patterns of social interaction in learning to write: some effects of network technologies. *Written Communication, 8*, pp. 79-113.

Harvey, D. (1990). *The condition of postmodernity: an enquiry into the origins of cultural change.* Oxford, U.K: Blackwell.

Hayek, F. (1945, September). The use of knowledge in society. American Economic Review, *35:4*, pp. 519-30.

Heim, M. (1991). The erotic ontology of cyberspace. In M. Benedikt (ed.), *Cyberspace: First Steps*, pp. 59-80. Boston, MA: MIT Press.

Hewitt, J. & Scardamalia, M. (1996, April). Design principles for the support of distributed processes. Paper presented at the Annual meeting of the American Educational Research Association, New York City. Retrieved January14th, 2000 from http://twilight.oise.utoronto.ca/abstracts/distributed.

Hickey, D. & Reiss, D. (1997, August 26). Cybersimulations: low-tech variations of high-tech applications for learning communities. Retrieved July 1st, 1999 from http://www.tc.cc.va.us/faculty/tcreisd/resource/cybrsims.htm.

Hillocks, G. (1995). *Teaching writing as reflective practice.* New York, NY: Teachers College Press.

Hofstader, D. (1979). *Gödel, Escher, Bach: An eternal golden braid.* New York, NY: Basic Books.

Hutchins, E. (1995). *Cognition in the wild.* Cambridge, MA: MIT Press.

Isaacs, E. A., Tang, J. C., & Morris, T. (1996). Piazza: a desktop environment supporting impromptu and planned interactions. In *Proceedings of CSCW '96*, pp. 315-328. New York, NY: Association for Computing Machinery.

Jameson, F. (1991). *Postmodernity, or the cultural logic of late capitalism.* Durham, NC: Duke University Press.

Jansson, E. (1999, December 12). Personal interview. Conducted at Stanford University, Palo Alto, California.

Johnson, D. W., Johnson, R. T., Holubec, E. J., & Roy, P. (1984). *Circles of learning: cooperation in the classroom.* Alexandria, VA: Association for Supervision and Curriculum Development.

Johnson, S. (1997). *Interface culture: how new technology transforms the way we create and communicate.* New York, NY: HarperCollins.

Kahn, P. H., Jr., & Friedman, B. (1998). Control and power in educational computing. In H. Bromley & M. W. Apple (eds.), *Education/technology/power: educational computing as a social practice.* Albany, NY: SUNY Press.

Kahn, T. M. (n.d.) Social ' know-who' for building virtual communities. *Educational Object Economy (EOE) Feature of the Week.* Retrieved August 21st, 1999 from http://www.eoe.org.

Kahn, T. M. (1999). Designing virtual communities for creativity and learning. *Edutopia, 6.2.* Retrieved August 21st, 1999 from http://glef.org/edutopia/newsletters/6.2/kahn.html.

Kauffman, S. (1996). *At home in the universe: the search for laws of self-organization and complexity.* Oxford, UK: Oxford University Press.

Kelly, K. (1995). *Out of control: the new biology of machines, social systems and the economic world.* Reading, MA: Addison-Wesley.

Kelly, S., & Allison, M. A., (1999). *The complexity advantage: how the science of complexity can help your business achieve peak performance.* New York, NY: McGraw-Hill.

Kemp, F. (1996). The beginnings of CMC-based instruction in Texas. Retrieved March 6th, 1997 from *http://english.ttu.edu/courses/5365/kemp/su96/support/lectures/enfi.texas.htm.*

Kemp, F. (1993). The origins of ENFI, network theory, and computer-based collaborative writing instruction at the University of Texas. In B. Bruce, J. Peyton, & T. Batson (eds.), *Network-based classrooms*, pp. 161-180. Cambridge, U.K.: Cambridge University Press.

Kiesler, S., Siegel, J. &. McGuire, T. W. (1988). Social psychological aspects of computer-mediated communication. In I. Greif (ed.), *Computer-supported cooperative work: a book of readings*. San Mateo, CA: Morgan Kaufmann Publishers, Inc.

Kleiner, A. & Roth, G. (1998). How to make experience your company's best teacher. In *Harvard Business Review on Knowledge Management*, pp. 137-151. Cambridge, Massachusetts: Harvard Business School Press.

Koch, M. (1998, Sept. 25). Call for papers on implementing tailorability in groupware. Email message originating from kochm@informatik.tu-muenchen.de.

Kohn, A. (1993). *Punished by rewards: the trouble with gold stars, incentive plans, A's, praise, and other bribes*. New York, NY: Houghton Mifflin.

Koschmann, T. (1996). Paradigm shifts and instructional technology: an introduction. In *CSCL: Theory and Practice of an Emerging Paradigm*, pp. 1-23. Mahwah, NJ: Lawrence Erlbaum Associates.

Koschmann, T., Kelson, A. C., Feltovich, P. J. & Barrows, H. S. (1996). Computer-supported problem-based learning: a principled approach to the use of computers in collaborative learning. In T. Koschmann (ed.), *CSCL: theory and practice of an emerging paradigm,* pp. 83-124. Mahwah, NJ: Lawrence Erlbaum Associates.

Kuhn, T. (1970). *The structure of scientific revolutions.* Chicago, IL: The University of Chicago Press.

Kvan, T., Wan, H. Y., & Vera, A. (1999). Supporting design studio learning: an investigation into design communication in computer-supported collaboration. In C. Hoadley (ed.), *Proceedings of CSCL '99,* pp. 328-332. New York, NY: Association for Computing Machinery.

Landow, G. P. (1997). *Hypertext 2.0: the convergence of contemporary critical theory and technology.* Baltimore, MD: Johns Hopkins University Press.

Lanham, R. A. (1993). *The electronic word: democracy, technology, and the arts.* Chicago, IL: The University of Chicago Press.

Laurel, B. (1993). *Computers as theatre.* Reading, MA: Addison-Wesley.

Lave, J. & Wenger, E. (1991). *Situated learning: legitimate peripheral participation.* Cambridge, MA: Cambridge University Press.

Lave, J. (1996). The practice of learning. In S. Chaiklin & J. Lave (eds.) *Understanding practice: perspectives on activity and context,* pp. 3-34. Cambridge, UK: Cambridge University Press.

Lay, M. M. (1992). The androgynous collaborator: the impact of gender studies of collaboration. In J. Forman (ed.), *New visions of collaborative writing*, pp. 82-104. Portsmouth, NH: Heinemann.

Lemke, J.L. (1996). Hypermedia and higher education. In T. M. Harrison & T. Stephen (eds.), *Computer networking and scholarly communication in the twenty-first-century university*, pp. 215-231. Albany, NY: SUNY Press.

Llanos, M. (1999, February 22). Teach the teachers technology. *ZDNN Tech News.* Retrieved February 22, 1999 from http://www.zdnet.com/zdnn/stories/news/0,4586,2214005,00.html.

Locker, K. O. (1992). What makes a collaborative writing team successful? A case study of lawyers and social workers in a state agency. In J. Forman (ed.), *New Visions of Collaborative Writing*. Portsmouth, NH: Heinemann.

Lynch, M. (1995). Laboratory space and the technological complex: an investigation of topical contexures. In S. L. Star (ed.), *Ecologies of knowledge: work and politics in science and technology.* Albany, NY: SUNY Press.

Markus, M. L. & Connolly, T. (1990). Why CSCW applications fail: problems in the adoption of interdependent work tools. In *Proceedings of CSCW '90,* pp. 371-380. New York, NY: Association for Computing Machinery, 1990.

Marquardt, M. (1997). *Action learning.* Alexandria, Virginia: American Society for Training and Development.

Marshak, R. (1996). Young and Rubicam: improving productivity with workflow. In P. Lloyd & R. Whitehead (eds.), *Transforming Organizations Through*

Groupware: Lotus Notes in Action, pp. 188-197. Berlin, Heidelberg, New York: Springer-Verlag,

Masumian, B. (1997, March.) Personal interview. Conducted at Advanced Micro Devices (AMD), Austin, Texas.

Maturana, H.R. & Varela, F.G. (1980). *Autopoiesis and cognition: the realization of the living.* Dordrecht, Holland; Boston: D. Reidel Pub. Co..

McCloud, S. (1993). *Understanding comics.* New York, NY: HarperCollins.

Michaelsen, L.K. & Black, R.H. (n.d.) *Building learning teams: the key to harnessing the power of small groups in higher education.* Retrieved September 10th, 1998 from http://eminfo.emc.maricopa.edu/innovation/ccl/models/team.html.

Michaelsen, L. K., Jones, C. F., & Watson, W.E. (n.d.) Beyond groups and cooperation: building high-performance learning teams. Retrieved September 10th, 1998 from http://eminfo.emc.maricopa.edu/innovation/ccl/models/beyond.html.

Mead, R. (2000, November 13). You've got blog. *New Yorker*, pp. 102-108

Moran, C. (1998, April). From a high-tech to a low-tech writing classroom: 'You can't go home again.' *Computers and Composition, 15:1.* Available http://www.cwrl.utexas.edu/~ccjrnl/Archives/v15/15_1_html/15_1_Feature.html.

Morningstar, C. & Farmer, F.R. (1991, 1994). The lessons of Lucasfilm's 'Habitat'. In M. Benedikt (ed.), *Cyberspace: First Steps*, pp. 273-301. Cambridge, MA: MIT Press. Retrieved on December 6, 2000 from http://imaginaryrealities.imaginary.com/volume2/issue9/lessons.html.

Morrison, D. & Goldberg, B. (1996). New actors, new connections: the role of local information infrastructures in school reform. In T. Koschmann (ed.), *CSCL: Theory and Practice of an Emerging Paradigm*, pp. 125-141. Mahwah, NJ: Lawrence Erlbaum Associates.

Murray, J.H. (1997). *Hamlet on the holodeck: the future of narrative in cyberspace.* Cambridge, MA: MIT Press.

Nakanishi, H., Yoshida, C., Nishimura, T., & Ishida, T. (1996). Freewalk: supporting casual meetings in a network. In *Proceedings of CSCW* '96, pp. 308-314. New York: Association for Computing Machinery.

Nardi, B.A. & O'Day, V.L. (1999). *Information ecologies: using technology with heart*. Cambridge, MA: MIT Press.

Neal, L. (1997). Virtual classrooms and communities. In S. Hayne, & W. Prinz (eds.), *Proceedings of GROUP '97*, pp. 81-90. New York: Association for Computing Machinery.

Neilson, R. E. (1997). *Collaborative technologies and organizational learning*. Hershey, Pa: Idea Group.

Neuwirth, C. M. & Wojahn, P. G. (1996). Learning to write: computer support for a cooperative process. In *CSCL: Theory and Practice of an Emerging Paradigm*, pp. 147-170. Mahwah, NJ: Lawrence Erlbaum Associates.

Noble, D.F. (1998). Digital diploma mills: the automation of higher education. *First Monday* 3.1 (January 1998). Retrieved September 1st, 1999 from http://www.firstmonday.dk/issues/issue3_1/noble/index.html.

Nonaka, I. (1998). The knowledge-creating company. In *Harvard Business Review on Knowledge Management*, pp. 21-46. Boston: Harvard Business School Publishing.

Norton Connect (n.d.) Student comments on Norton Connect. Retrieved June 27th, 2000 from http://www.wwnorton.com/connect/comments/Class1.htm.

Novak, M. (1994). Liquid architectures in cyberspace. In M. Benedikt (ed.), *Cyberspace: First Steps*, pp. 225-254. Cambridge, MA: MIT Press.

O'Dell, C. & Grayson, C. J. Jr. (1998). *If only we knew what we know: the transfer of internal knowledge and best practice.* New York: The Free Press.

Orlikowski, W. (1996). Evolving with notes: organizational change around groupware technology. *In Groupware and Teamwork: Invisible Aid or Technical Hindrance?*, pp. 23-59. West Sussex, England: John Wiley & Sons.

Pirsig, R. (1984). *Zen and the art of motorcycle maintenance: an inquiry into values.* New York: Bantam Books.

Plumtree Software. *Plumtree Corporate Portal 3.0.* San Francisco: Plumtree Software. Retrieved January 8th, 2000 from http://www.plumtree.com.

Polanyi, M. (1997). Tacit knowledge. In L. Prusak (ed.), *Knowledge in Organizations.* Boston: Butterworth-Heinemann.

Rawlins, G. J. E. (1998). *Slaves of the machine: the quickening of computer technology.* Cambridge, MA: MIT Press.

Regan, A. (1993). 'Type normal like the rest of us': writing, power, and homophobia in the networked composition classroom. *Computers and Composition: An International Journal for Teachers of Writing, 10.4.* pp. 11-23.

Reimer, T. G. & Edelson, D. C. (1999). Using a CSCL tool to support a legislative role-playing simulation: hopes, fears, and challenges. In C. Hoadley (ed.), *Proceedings of CSCL '99*, pp. 466-474. New York, NY: Association for Computing Machinery.

Repenning, A., Ioannidou, A., & Phillips, J. (1999). Collaborative use and design of interactive simulations. In C. Hoadley (ed.), *Proceedings of CSCL '99*, pp. 475-487. New York, NY: Association for Computing Machinery.

Mitchel, R. (1994). *Turtles, termites, and traffic jams: explorations in massively parallel Microworlds.* Cambridge, MA: MIT Press.

Roschelle, J. (1996). Learning by collaborating: converging conceptual change. In *CSCL: Theory and Practice of an Emerging Paradigm (Computers, Cognition, and Work*), pp. 209-248. Mahwah, NJ: Lawrence Erlbaum Associates.

Roschelle, J., & Teasley, S. (1995). The construction of shared knowledge in collaborative problem solving. In C. E. O'Malley (ed.), *Computer Supported Collaborative Learning*. Heidelberg: Springer-Verlag.

Roschelle, J., DiGiano, C., Pea, R., & Kaput, J. (n.d.) Educational software components of tomorrow (ESCOT). Retrieved September 18th, 2000 from http://www.escot.org/External/MSET_ESCOT.html.

Roskelly, H. (1994). The risky business of group work. In G. Tate, E.P.J. Corbett, & N. Myers (eds.), *The Writing Teacher's Sourcebook*, 3rd ed., pp. 141-146. New York, Oxford: Oxford University Press.

Ryan, H.W. (1995). Building computing solutions with the team metaphor. In D. Coleman & R. Khanna (eds.), *Groupware: Technology and Applications*, pp. 477-495. Upper Saddle River, NJ: Prentice Hall.

Scardamalia, M. & Bereiter, C. (1995). Computer support for knowledge building communities. In *CSCL: Theory and Practice of an Emerging Paradigm*, pp. 249-268. Mahwah, NJ: Lawrence Erlbaum Associates.

Schank, R. (1994). The effect of grades. In *Engines for Education*. Evanston: The Institute for the Learning Sciences. Retrieved March 1st, 1996 from http://www.ils.nwu.edu/~e_for_e/nodes/NODE-56-pg.html.

Schneier, L. (1995). The implementation of enterprise-ware at the World Bank: a case study. In D. Coleman & R. Khanna (eds.), *Groupware*: *Technology and Applications*, pp. 378-407. Upper Saddle River, NJ: Prentice Hall.

Scholes, R. (1985). *Textual power: literary theory and the teaching of English*. New Haven: Yale University Press.

Schwarz, R. M. (1994). *The skilled facilitator: practical wisdom for developing effective groups*. San Francisco: Jossey-Bass.

Selfe, C. (1992). Computer-based conversations and the changing nature of collaboration. In J. Forman (ed.), *New Visions of Collaborative Writing*, pp. 147-169. Portsmouth, NH: Boynton/Cook Heinemann.

Selfe, C. (1992). Preparing English teachers for the virtual age: the case for technology critics. In G. E. Hawisher & P. LeBlanc (eds.), *Re-Imagining Computers and Composition*: *Teaching and Research in the Virtual Age*, pp. 24-42. Portsmouth, NH: Boynton/Cook.

Selfe, C. & Selfe, R., (1994). The politics of the interface: power and its exercise in electronic contact zones. In *College Composition and Communication*,

4(45), pp. 480-504. Retrieved May 29th, 1999 from http://www.hu.mtu.edu/~cyselfe/cindypages/texts/politics.html.

Selfe, C. (1996). Theorizing e-mail for the practice, instruction, and study of literacy. In P. Sullivan & J. Dautermann (eds.), *Electronic Literacies in the Workplace*, pp. 255- 293. Urbana, IL and Houghton, MI: National Council of Teachers of English and Computers and Composition Press.

Selfe, C. & Hawisher, G. (1997). Wedding the technologies of writing portfolios and computers: the challenges of electronic classroom. In K. B. Yancey & I. Wiser (eds.), *Situating Portfolios: Four Perspectives*, pp. 305-321. Logan, UT: Utah State University Press.

Senge, P. (1990). *The fifth discipline: the art and practice of building a learning organization*. New York: Doubleday.

Slatin, J. (1992). Is there a class in this text? Creating knowledge in the electronic classroom. In E. Barrett (ed.), *Sociomedia: Multimedia, Hypermedia, and the Social Construction of Knowledge*, pp. 27-51. Cambridge, Mass: MIT Press.

Smith, R. B., Sipusic, M. J., & Pannoni, R. L. Experiments comparing face-to-face with virtual collaborative learning. In C. Hoadley (ed.), *Proceedings of CSCL '99*, pp. 558-566. New York, NY: Association for Computing Machinery.

Smolin, L. (1997). *The life of the cosmos*. New York: Oxford University Press.

Sommers, E. (1992). Political impediments to virtual reality. In Hawisher, G. E., & LeBlanc, P. (eds.), *Re-imagining computers and composition: teaching and research in the virtual age*, pp. 43-57. Portsmouth, NH: Boynton/Cook.

Spellman, P. J., Mosier, J. N., Deus, L. M. & Carlson, J. A. (1997). Collaborative virtual workspace. In Hayne, S. C. & Prinz, W. (eds.), *Proceedings of GROUP '97*, pp. 197-203. New York: Association for Computing Machinery.

Spohrer, J. (n.d.) A brief note on open source communities and intellectual capital appreciation licenses. Educational Object Economy (EOE) Web site. Retrieved June 16th, 1999 from http://www.eoe.org.

Sproull, L. & Kiesler, S. (eds.) (1987). *Computing and change on campus.* New York: Cambridge University Press.

Sproull, L. & Kiesler, S. (1991). *Connections: new ways of working in the networked organization.* Cambridge, MA: MIT Press.

Stone, R. A. (1995). *The war of desire and technology at the close of the mechanical age.* Cambridge, MA: MIT Press.

Suchman, L. (1987). *Plans as situated actions: the problems of human-machine communication.* Cambridge: Cambridge University Press.

Syverson, M. A. (1999). *The wealth of reality: an ecology of composition.* Carbondale, IL: Southern Illinois University Press.

Trimbur, J. (1989). Consensus and difference in collaborative learning. *College English, 51*, pp. 602-16.

Trimbur, J., & L. A. Braun (1992). Laboratory life and the determination of authorship. In J. Forman (ed.), *New visions of collaborative writing.* Portsmouth: Heinemann.

Tollmar, K., Sandor, O., & Schömer, A. (1996). Supporting social awareness @work: design and experience. In *Proceedings of CSCW '96*, pp. 298-307. New York: Association for Computing Machinery.

Turkle, S. (1995). *Life on the screen: identity in the age of the internet.* New York: Simon & Schuster.

Ullman, E. (1997). *Close to the machine: technophilia and its discontents.* San Francisco: City Lights Books.

Underhill, P. (1999). *Why we buy: the science of shopping*. New York: Simon & Schuster.

Varn, R. J. (1999, February). Higher education – from industrial age to information age education. *Converge Magazine*, *2.2*, pp. 74-75.

Vygotsky, L. S. (1978). *Mind in society: the development of higher psychological processes.* Cambridge, U.K: Cambridge University Press.

Waldrop, M. M. (1992). *Complexity : the emerging science at the edge of order and chaos.* New York: Touchstone (Simon & Schuster).

Warshauer, S. (1995). Rethinking teacher authority to counteract homophobic prejudice in the networked classroom: a model of teacher response and overview of classroom methods. *Computers and Composition: An International Journal for Teachers of Writing 12.1*, pp. 97-111.

Warschauer, M. (1999). *Electronic literacies: language, culture, and power in on-line education.* Mahwah, NJ: Lawrence Erlbaum Associates.

WebCT (2000). WebCT. Retrieved October 4th, 2000 from http://www.webct.com.

Weizenbaum, J. (1976). *Computer power and human reason: from judgment to calculation.* San Francisco: W. H. Freeman.

Weiner, R. S. (2000, June 21). Instructors say online courses involve more work at same pay. *New York Times*. Retrieved June 21st, 2000 from http://www.nytimes.com/library/tech/00/06/cyber/education/21education.html.

Wiener, H. S. (1994). Collaborative learning in the classroom: a guide to evaluation. In Tate, G., Corbett, E.P.J., & Myers, N. (eds.) *The writing*

teacher's sourcebook, 3rd ed., pp. 132-140. New York: Oxford University Press.

Winograd, T. (1988). A language/action perspective on the design of cooperative work. In I. Greif (ed.), *Computer-supported cooperative work: a book of readings*. San Mateo, CA: Morgan Kaufmann.

Winograd, T. and Flores, F. (1987). *Understanding computers and cognition: a new foundation for design.* Norwood, NJ: Ablex.

Wolfe, J. (1999). Why do women feel ignored? Gender differences in computer-mediated interactions. *Computers & Composition, 16*, pp. 153-166.

Wolfe, J. (2000). Gender, ethnicity and classroom discourse: communication patterns of hispanic and white students in networked classrooms. *Written Communication, 17*, 491-519.

Zeni, J. (1994). Literacy, technology, and teacher education. In C. Selfe & S. Hilligoss (eds.), *Literacy and computers: the complications of teaching and learning with technology*, pp. 76-86. New York: Modern Language Association.

Zuboff, S. (1988). *In the age of the smart machine: the future of work and power.* New York: Basic Books.

The vita has been removed from the digitized version of this document.