
Collaboration Technology for Communities

In the age of information-overload, lifelong learning and collaboration are essential aspects of most innovative work. Fortunately, the computer technology that drives the information explosion also has the potential to help individuals and groups learn, on demand, much of what they need to know. In particular, applications on the Internet can be designed to capture knowledge as it is generated within a community of practice and to deliver relevant knowledge when it is useful.

Computer-based design environments for skilled domain workers have recently graduated from research prototypes to commercial products, supporting the learning of individual designers. Such systems do not, however, adequately support the collaborative nature of work or the evolution of knowledge within communities of practice. If innovation is to be supported within collaborative efforts, these domain-oriented design environments (DODEs) must be extended to become collaborative information environments (CIEs), capable of providing effective community memories for managing information and learning within constantly evolving collaborative contexts. In particular, CIEs must provide functionality that facilitates the construction of new knowledge and the shared understanding necessary to use this knowledge effectively within communities of practice.

This chapter reviews three stages of work on artificial (computer-based and Web-based) systems that augment the intelligence of people and organizations. NetSuite illustrates the DODE approach to supporting the work of individual designers with learning-on-demand. WebNet extends this model to CIEs that support collaborative learning by groups of designers. Finally, WebGuide shows how a computational perspectives mechanism for CIEs can support the construction of knowledge and of shared understanding within groups. According to recent theories of cognition, human intelligence is the product of tool use and of social mediations as well as of biological development; CIEs are designed to enhance this intelligence by providing computationally powerful tools that are supportive of social relations.

Thereby, this chapter carries out a transition from systems that use AI techniques and computational power to computer-based media that support communication and collaboration. In part, this is a difference of emphasis, as the media may still incorporate significant computation. However, it is also a shift in the locus of intelligence from clever software to human group cognition.

1. Introduction: The Need for Computer Support of Lifelong Collaborative Learning

The creation of innovative artifacts and helpful knowledge in our complex world—with its refined division of labor and its flood of information—requires continual learning

and collaboration. Learning can no longer be conceived of as an activity confined to the classroom and to an individual's early years. Learning must continue while one is engaged with other people as a worker, a citizen and an adult learner for many reasons:

- Innovative tasks are ill-defined; their solution involves continual learning and the creative construction of knowledge whose need could not have been foreseen (Rittel & Webber, 1984).
- There is too much knowledge, even within specific subject areas, for anyone to master it all in advance or on one's own (Zuboff, 1988).
- The knowledge in many domains evolves rapidly and often depends upon the context of one's task situation, including one's support community (Senge, 1990).
- Frequently, the most important information has to do with a work group's own structure and history, its standard practices and roles and the details and design rationale of its local accomplishments (Orr, 1990).
- People's careers and self-directed interests require various new forms of learning at different stages as their roles in communities change (Argyris & Schön, 1978).
- Learning—especially collaborative learning—has become a new form of labor, an integral component of work and organizations (Lave & Wenger, 1991).
- Individual memory, attention and understanding are too limited for today's complex tasks; divisions of labor are constantly shifting, and learning is required to coordinate and respond to the changing demands on community members (Brown & Duguid, 1991).
- Learning necessarily includes organizational learning: social processes that involve shared understandings across groups. These fragile understandings are both reliant upon and in tension with individual learning, although they can also function as the cultural origin of individual comprehension (Vygotsky, 1930/1978).

The pressure on individuals and groups to continually construct new knowledge out of massive sources of information strains the abilities of unaided human cognition. Carefully designed computer software promises to enhance the ability of communities to construct, organize and share knowledge by supporting these processes. However, the design of such software remains an open research area.

The contemporary need to extend the learning process from schooling into organizational and community realms is known as *lifelong learning*. Our past research at the University of Colorado's Center for LifeLong Learning and Design explored the computer support of lifelong learning with what we call *domain-oriented design environments* (DODEs). This chapter argues for extending that approach to support work within communities of practice with what it will term *collaborative information environments* (CIEs) applied both to design tasks and to the construction of shared knowledge. This chapter illustrates three stages that our efforts with illustrative software systems have evolved through during the 1990s.

Section 2 of this chapter highlights how computer support for lifelong learning has already been developed for individuals such as designers. It argues, however, that DODEs—such as the commercial product *NetSuite*—that deliver domain knowledge to individuals when it is relevant to their task are not sufficient for supporting innovative work within collaborative communities. Section 3 sketches a theory of how software productivity environments for design work by individuals can be extended to support organizational learning in collaborative work structures known as communities of

practice; a scenario of a prototype system called WebNet illustrates this. Section 4 of this chapter discusses the need for mechanisms within CIEs to help community members construct knowledge in their own personal perspectives while also negotiating shared understanding about evolving community knowledge; this is illustrated by the perspectives mechanism in WebGuide, discussed in terms of three learning applications. A concluding section locates this discussion within the context of broader trends in computer science.

2. Augmenting the Work of Individual Designers

In this section I discuss how our DODE approach, which has now emerged in commercial products, provides support for individual designers. However, because design (such as the layout, configuration and maintenance of computer networks) now typically takes place within communities of practice, it is desirable to provide computer support at the level of these communities as well as at the individual designer's level and to include local community knowledge as well as domain knowledge. Note that much of what is described in this section about our DODE systems applies to a broad family of design critiquing systems developed by others for domains such as medicine (Miller, 1986), civil engineering (Fu, Hayes, & East, 1997) and software development (Robbins & Redmiles, 1998).

2.1 Domain-Oriented Design Environments

Many innovative work tasks can be conceived of as *design* processes: elaborating a new idea, planning a presentation, balancing conflicting proposals or writing a visionary report, for example. While designing can proceed on an intuitive level based on tacit expertise, it periodically encounters breakdowns in understanding where explicit reflection on new knowledge may be needed (Schön, 1983). Thereby, designing entails learning.

For the past decade, we have explored the creation of DODEs to support workers as designers. These systems are *domain-oriented*: they incorporate knowledge specific to the work domain. They are able to recognize when certain breakdowns in understanding have occurred and can respond to them with appropriate information (Fischer *et al.*, 1993). They support learning-on-demand.

To go beyond the power of pencil-and-paper representations, software systems for lifelong learning must "understand" something of the tasks they are supporting. This is accomplished by building knowledge of the domain into the system, including capturing design objects and design rationale. A DODE typically provides a computational workspace within which a designer can construct an artifact and represent components of the artifact being constructed. Unlike a CAD system, in which the software only stores positions of lines, a DODE maintains a *representation* of objects that are meaningful in the domain. For instance, an environment for local-area network (LAN) design (a primary example in this chapter) allows a designer to construct a network's design by selecting items from a palette representing workstations, servers, routers, cables and other devices from the LAN domain, and configuring these items into a system design. Information about each device is represented in the system.

A DODE can contain domain knowledge about constraints, rules of thumb and design rationale. It uses this information to respond to a current design state with active advice. Our systems use a mechanism we call *critiquing* (Fischer *et al.*, 1998). The system maintains a representation of the semantics of the design situation: usually the two-dimensional location of palette items representing design components. Critic rules are applied to the design representation; when a rule “fires,” it posts a message alerting the designer that a problem might exist. The message includes links to information such as design rationale associated with the critic rule.

For instance, a LAN DODE might notice that the length of a cable in a design exceeds the specifications for that type of cable; that a router is needed to connect two subnets; or that two connected devices are incompatible. At this point, the system could signal a possible design breakdown and provide domain knowledge relevant to the cited problem. The evaluation of the situation and the choice of action is up to the human designer, but now the designer has been given access to information relevant to making a decision (Fischer *et al.*, 1996).

2.2 NetSuite: A Commercial Product

Many of the ideas in our DODEs are now appearing in commercial products, independently of our efforts. In particular, there are several environments for designing LANs. As an example, consider NetSuite, a highly rated system that illustrates current best practices in LAN design support. This is a high-functionality system for skilled domain professionals who are willing to make the effort required to learn to use its rich set of capabilities (see Figure 5-1). NetSuite contains a wealth of domain knowledge. Its palette of devices, which can be placed in the construction area, numbers over 5,000, with more available for download from the vendor every month. Each device has associated parameters defining its characteristics, limitations and compatibilities—domain knowledge used by the critics that validate designs.

Figure 5-1 goes approximately here

In NetSuite, one designs a LAN from scratch, placing devices and cables from the palette. As the design progresses, the system validates it, critiquing it according to rules and parameters stored in its domain knowledge. The designer is informed about relevant issues in a number of ways: lists of devices to substitute into a design are restricted by the system to compatible choices, limited design rationale is displayed with the option of linking to further details and technical terms are defined with hypertext links. In addition to the construction area, there are LAN tools, such as an automated IP address generator and utilities for reporting on physically existing LAN configurations. When a design is completed, a bill-of-materials can be printed out and an HTML page of it can be produced for display on the Internet. NetSuite is a knowledgeable, well-constructed system to support an individual LAN designer.

2.3 The Need to Go Further

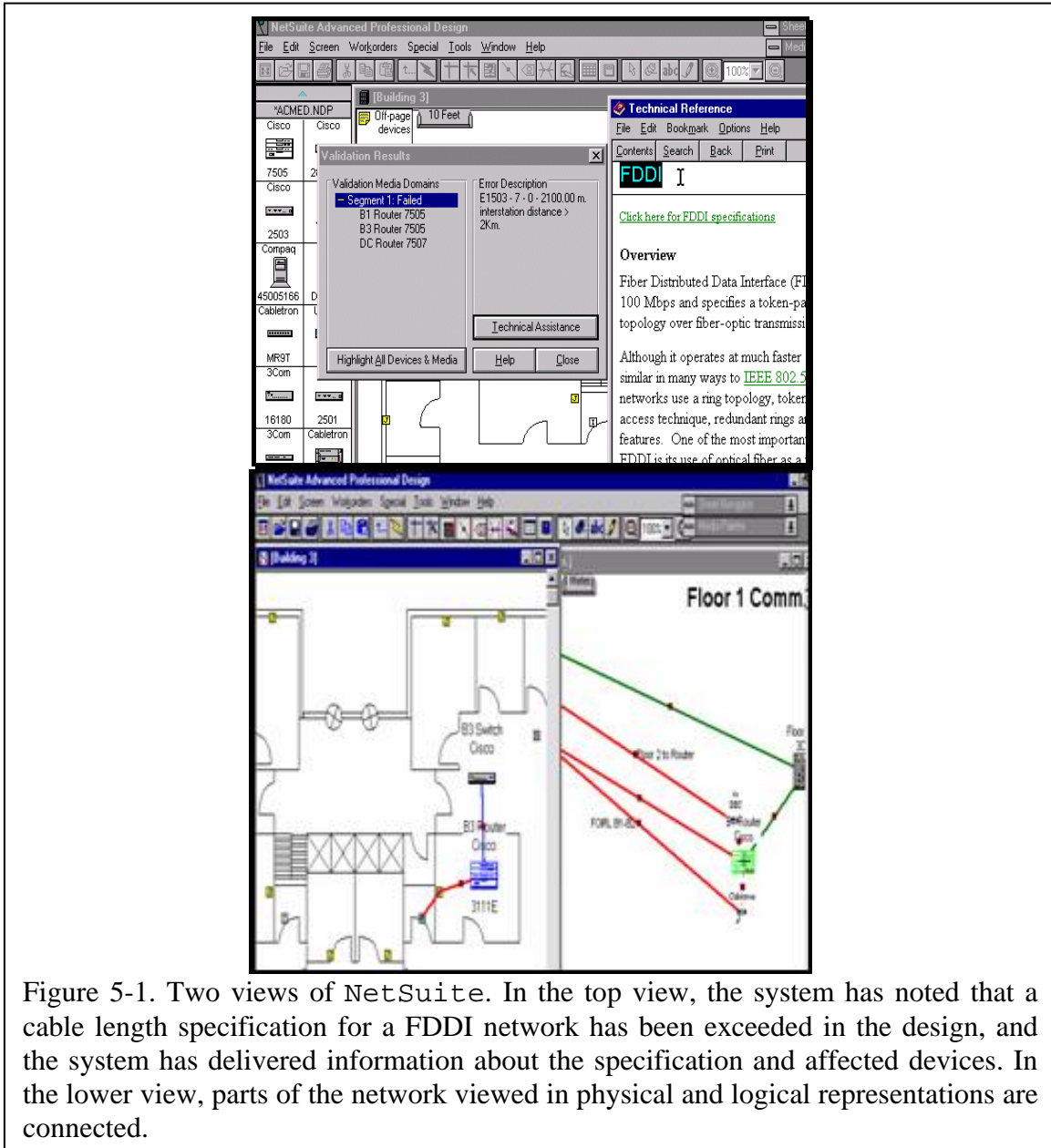


Figure 5-1. Two views of NetSuite. In the top view, the system has noted that a cable length specification for a FDDI network has been exceeded in the design, and the system has delivered information about the specification and affected devices. In the lower view, parts of the network viewed in physical and logical representations are connected.

Based on our understanding of organizational learning and our investigation of LAN design communities, we believe that in a domain like LAN management no closed system will suffice. The domain knowledge required to go beyond the functionality of NetSuite is too open-ended, too constantly changing and too dependent upon local circumstances. The next generation of commercial DODEs will have to support extensibility by end-users and collaboration within communities of practice. While a system like NetSuite has its place in helping to design complex networks from scratch, most work of LAN managers involves extending existing networks, debugging breakdowns in service and planning for future technologies.

Many LAN management organizations rely on home-grown information systems because they believe that critical parts of their local information are unique. Each community of practice has its own ways of doing things. Generally, these local practices are understood tacitly and are propagated through apprenticeship (Lave & Wenger, 1991). This causes problems when the old-timer who set things up is gone and when a newcomer does not know who to ask or even what to ask. A community memory is needed that captures local knowledge when it is generated (e.g., when a device is configured) and delivers knowledge when it is needed (when there is a problem with that device) without being explicitly queried.

The burden of entering all this information in the system must be distributed among the people doing the work and must be supported computationally to minimize the effort required. This means:

- The DODE knowledge base should be integrated with work practices in ways that capture knowledge as it is created.
- The benefits of maintaining the knowledge base have to be clearly experienced by participants.
- There may need to be an accepted distribution of roles related to the functioning of the organizational memory.
- The software environment must be thoroughly interactive so that users can easily enter data and comments.
- The information base should be seeded with basic domain knowledge so that users do not have to enter everything and so that the system is useful from the start.
- As the information space grows, there should be ways for people to restructure it so that its organization and functionality keep pace with its evolving contents and uses (Fischer *et al.*, 1999).

DODEs must be extended in these ways to support communities of practice, and not just isolated designers. This reflects a shift of emphasis from technical domain knowledge to local, socially-based community knowledge.

3. Supporting Communities of Practice

In this section, I briefly define “community of practice”—a level of analysis increasingly important within discussions of computer-supported cooperative work (CSCW)—and suggest that these communities need group memories to carry on their work. The notion of DODEs must be extended to support the collaborative learning that needs to take place within these communities. A scenario demonstrates how a CIE prototype named *WebNet* can do this.

3.1 Community Memories

3.1.1 Communities of Practice

All work within a division of labor is social (Marx, 1867/1976). The job that one person performs is also performed similarly by others and relies upon vast social networks. That is, work is defined by social practices that are propagated through socialization, apprenticeship, training, schooling and culture (Bourdieu, 1972/1995;

Giddens, 1984b; Lave & Wenger, 1991), as well as by explicit standards. Often, work is performed by collaborating teams that form communities of practice within or across organizations (Brown & Duguid, 1991). These communities evolve their own styles of communication and expression, or genres (Bakhtin, 1986a; Yates & Orlikowski, 1992).

For instance, interviews we conducted showed that computer network managers in different departments at our university work in concert. They need to share information about what they have done and how it is done with other team members and with other LAN managers elsewhere. For such a community, information about their own situation and local terminology may be even more important than generic domain knowledge (Orr, 1990). Support for LAN managers must provide memory about how individual local devices have been configured, as well as offer domain knowledge about standards, protocols, compatibilities and naming conventions.

Communities of practice can be co-located within an organization (e.g., at our university) or across a discipline (e.g., all managers of university networks). Before the World Wide Web existed, most computer support for communities of practice targeted individuals with desktop applications. The knowledge in the systems was mostly static domain knowledge. With intranets and dynamic Web sites, it is now possible to support distributed communities and also to maintain interactive and evolving information about local circumstances and group history. Communities of practice need to be able to maintain their own memories. The problem of adoption of organizational memory technologies by specific communities involves complex social issues beyond the scope of this chapter. For a review of common adoption issues and positive and negative examples of responses, see (Grudin, 1990; Orlikowski, 1992; Orlikowski *et al.*, 1995).

3.1.2 Digital Memories for Communities of Practice

Human and social evolution can be viewed as the successive development of increasingly effective forms of memory for learning, storing and sharing knowledge. Biological evolution gave us episodic, mimetic and mythical memory; then cultural evolution provided oral and written (external and shared) memory; finally modern technological evolution generates digital (computer-based) and global (Internet-based) memories (Donald, 1991; Norman, 1993).

At each stage, the development of hardware capabilities must be followed by the definition and adoption of appropriate skills and practices before the potential of the new information technology can begin to be realized. External memories, incorporating symbolic representations, facilitated the growth of complex societies and sophisticated scientific understandings. Their effectiveness relied upon the spread of literacy and industrialization. Similarly, while the proliferation of networked computers ushers in the possibility of capturing new knowledge as it is produced within work groups and delivering relevant information on demand, the achievement of this potential requires the careful design of information systems, software interfaces and work practices. New computer-based organizational memories must be matched with new social structures that produce and reproduce patterns of organizational learning (Giddens, 1984b; Lave & Wenger, 1991).

Community memories are to communities of practice what human memories are to individuals. They embody organizational memory in external repositories that are accessible to community members. They make use of explicit, external, symbolic representations that allow for shared understanding within a community. They make

organizational learning possible within the group (Ackerman & McDonald, 1996; Argyris & Schön, 1978; Borghoff & Parechi, 1998; Buckingham Shum & Hammond, 1994; Senge, 1990).

3.1.3 Integrative Systems for Community Memory

Effective community memory relies on integration. Tools for representing design artifacts and other work tasks must be related to rich repositories of information that can be brought to bear when needed. Communication about artifacts under development should be tied to that artifact so they retain their context of significance and their association with each other. Also, members of the community of practice must be integrated with each other in ways that allow something one member learned in the past to be delivered to other members when they need it in the future. One model for such integration—on an individual level—is the human brain, which stores a wealth of memories over a lifetime of experience, thought and learning in a highly inter-related associative network that permits effective recall based on subjective relevance. This—and not the traditional model of computer memory as an array of independent bits of objective information—is the model that must be extended to community memories.

Of course, we want to implement community memories using computer memory. Perhaps the most important goal is integration, in order to allow the definition of associations and other inter-relationships. For instance, in a system using perspectives, like those to be discussed in section 4, it is necessary for all information to be uniformly structured with indications of perspective and linking relationships. A traditional way to integrate information in a computer system is with a relational database. This allows associations to be established among arbitrary data. It also provides mechanisms like SQL queries to retrieve information based on specifications in a rather comprehensive language. Integrating all the information of a design environment in a unified database makes it possible to build bridges from the current task representation to any other information. Certainly, object-oriented or hybrid databases and distributed systems that integrate data on multiple computers can provide the same advantages. Nor does an underlying query language like SQL have to be exposed to users; front-end interfaces can be much more graphical and domain-oriented (Buckingham Shum, 1998).

Communities themselves must also be integrated. The Web provides a convenient technology for integrating the members of a community of practice, even if they are physically dispersed or do not share a homogeneous computer platform. In particular, intranets are Web sites designed for communication within a specific community rather than world-wide. *WebNet*, for instance, is intranet-based software that we prototyped for LAN management communities. It includes a variety of communication media as well as community memory repositories and collaborative productivity tools. It will be discussed later in this section.

Dynamic Web pages can be interactive in the sense that they accept user inputs through selection buttons and text entry forms. Unlike most forms on the Web that only provide information (like product orders, customer preferences, or user demographics) to the webmaster, intranet feedback may be made immediately available to the user community that generated it. For instance, the *WebNet* scenario below includes an interactive glossary. When someone modifies a glossary definition, the new definition is displayed to anyone looking at the glossary. Community members can readily comment

on the definitions or change them. The history of the changes and comments made by the community is shared by the group. In this way, intranet technology can be used to build systems that are CIEs in which community members deposit knowledge as they acquire it so that other members can learn when they need or want to, and can communicate with others about their learning. This model illustrates computer support for collaborative learning with digital memories belonging to communities of practice.

3.2 Extending the DODE Approach to CIEs for Design

To provide computer support for collaborative learning with CIEs, we first have to understand the process of collaborative learning. Based on this analysis, we can see how to extend the basic characteristics of a DODE to create a CIE.

3.2.1 The Process of Collaborative Learning

The ability of designers to proceed based on their existing tacit expertise (Polanyi, 1962) periodically breaks down and they have to rebuild their understanding of the situation through explicit reflection (Schön, 1983). This reflective stage can be helped if they have good community support and effective computer support to bring relevant new information to bear on their problem. When they have comprehended the problem and incorporated the new understanding in their personal memories, we say they have learned. The process of *design* typically follows this cycle of breakdown and reinterpretation in learning (see Figure 5-2, cycle on left).

Figure 5-2 goes approximately here

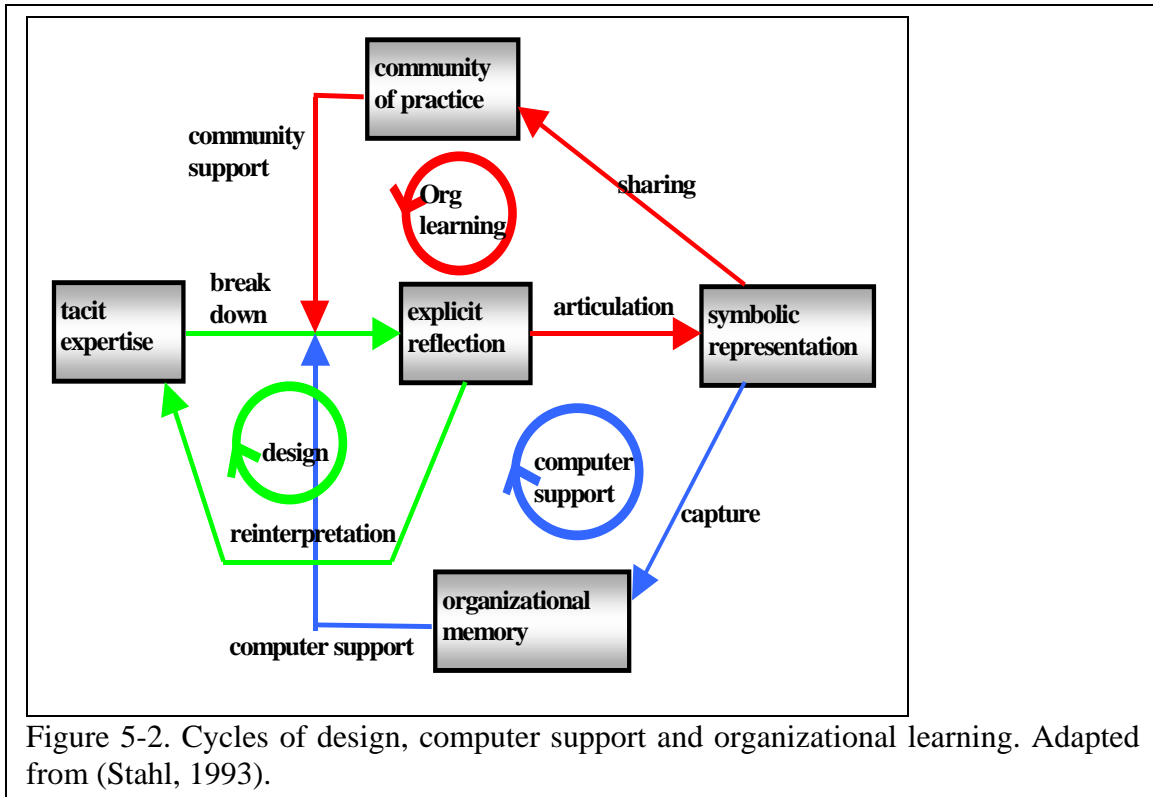


Figure 5-2. Cycles of design, computer support and organizational learning. Adapted from (Stahl, 1993).

When design tasks take place in a collaborative context, the reflection results in articulation of solutions in language or in other symbolic representations. The articulated new knowledge can be shared within the community of practice. Such knowledge, created by the community, can be used in future situations to help a member overcome a breakdown in understanding. This cycle of collaboration is called *organizational learning* (see Figure 5-2, upper cycle). The personal reflection and the collaborative articulation of shared perspectives interacting together make innovation possible (Boland & Tenkasi, 1995; Tomasello, Kruger, & Ratner, 1993).

Organizational learning can be supported by computer-based systems of organizational memory if the articulated knowledge is captured in a digital symbolic representation. The information must be stored and organized in a format that facilitates its subsequent identification and retrieval. In order to provide *computer support*, the software must be able to recognize breakdown situations when particular items of stored information might be useful to human reflection (see Figure 5-2, lower cycle). DODEs provide computer support for design by individuals. They need to be extended to collaborative information environments (CIEs) to support organizational learning in communities of practice.

3.2.2 Extending the DODE Approach to CIEs for Design

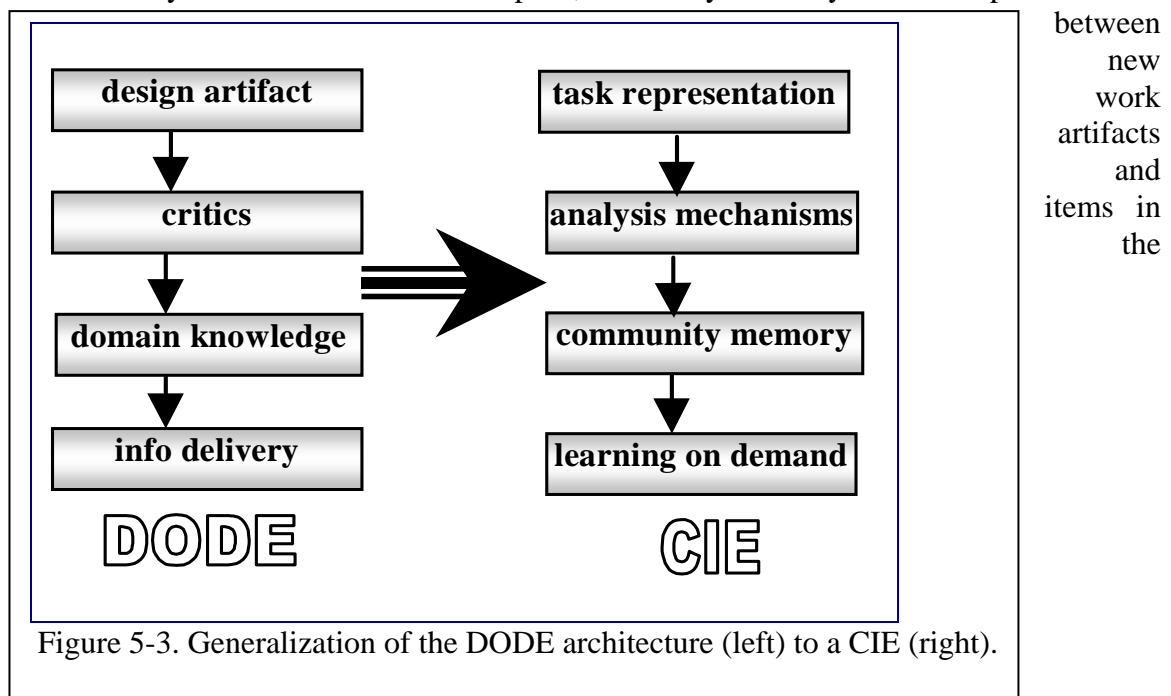
The key to active computer support that goes significantly beyond printed external memories is to have the system deliver the right information at the right time in the right way (Fischer *et al.*, 1998). To do this, the software must be able to analyze the state of the work being undertaken, identify likely breakdowns, locate relevant information and deliver that information in a timely manner.

Systems like NetSuite and our older prototypes used critics based on domain knowledge to deliver information relevant to the current state of a design artifact being constructed in the design environment work space (see Figure 5-3, left).

Figure 5-3 goes approximately here

One can generalize from the critiquing approach of these DODEs to arrive at an overall architecture for organizational memories. The core difference between a DODE and a CIE is that a DODE focuses on delivering domain knowledge, conceived of as relatively static and universal, while a CIE is built around forms of community memory, treated as constantly evolving and largely specific to a particular community of practice. Where DODEs relied heavily on a set of critic rules predefined as part of the domain knowledge, CIEs generalize the function of the critiquing mechanisms.

In a CIE, it is still necessary to maintain some representation of the task as a basis for the software to take action. This task representation plays the role of the design artifact in a DODE, triggering critics and generally defining the work context in order to decide what is relevant. This is most naturally accomplished if work is done within the software environment. For instance, if communication about designs takes place within the system where the design is constructed, then annotations and email messages can be linked directly to the design elements they discuss. This reduces problems of deixis (comments referring to “that” object “over there”). It also allows related items to be linked together automatically. In an information-rich space, there may be many relationships of interest



organizational memory. For instance, when a LAN manager debugs a network, links between network diagrams, topology designs, LAN diary entries, device tables and an interactive glossary of local terminology can be browsed to discover relevant information.

The general problem for a CIE is to define analysis mechanisms that can bridge the gap from task representation to relevant community memory information items in order to support learning on demand (see Figure 5-3, right).

To take a very different example, suppose a student is writing a paper within a software environment that includes a digital library of papers written by her and her colleagues. An analysis mechanism to support her learning might compare sentences or paragraphs in her draft (which functions as a task representation) to text from other papers and from email discussions (the community memory) to find excerpts of potential interest to her. We use latent semantic analysis (Landauer & Dumais, 1997) to mine our email repository (Lindstaedt & Schneider, 1997), and are exploring similar uses of this mechanism to link task representations to textual information to support organizational learning. Other retrieval mechanisms might be appropriate for mining catalogs of software agents or components, design elements and other sorts of organizational memories.

Using our example of LAN design, I next show how a CIE might function in this domain. I present a scenario of use of *WebNet*, a prototype I developed to extend our DODE concept to explicitly support communities of LAN designers.

3.3 *webNet*: Scenario of a CIE for Design

3.3.1 Critiquing and Information Delivery

Kay is a graduate student who works part-time to maintain her department's LAN. The department has a budget to extend its network and has asked Kay to come up with a design. Kay brings up *WebNet* in her Web browser. She opens up the design of her department's current LAN in the *LAN Design Environment*, an *Agentsheets* (Repenning, 1994) simulation applet. Kay starts to add a new subnet. Noticing that there is no icon for an Iris graphics workstation in her palette, Kay selects the *WebNet* menu item for the *Simulations Repository Web* page (see Figure 5-4, left frame). This opens a Web site that contains simulation agents that other *Agentsheets* users have programmed. *WebNet* opens the repository to display agents that are appropriate for *WebNet* simulations. Kay locates a simulation agent that someone else has created with the behavior of an Iris workstation. She adds this to her palette and to her design.

Figure 5-4 goes approximately here

When Kay runs the LAN simulation, *WebNet* proactively inserts a router (see Figure 5-4, upper right) and informs Kay that a router is needed at the intersection of the two subnets. *WebNet* displays some basic information about routers and suggests several Web sites with details about different routers from commercial vendors (see Figure 5-4, lower right). Here, *WebNet* has signaled a breakdown in Kay's designing and provided easy access to sources of information for her to learn what she needs to know on demand. This information includes generic domain knowledge like definitions of technical terms, current equipment details like costs and community memory from related historical emails.

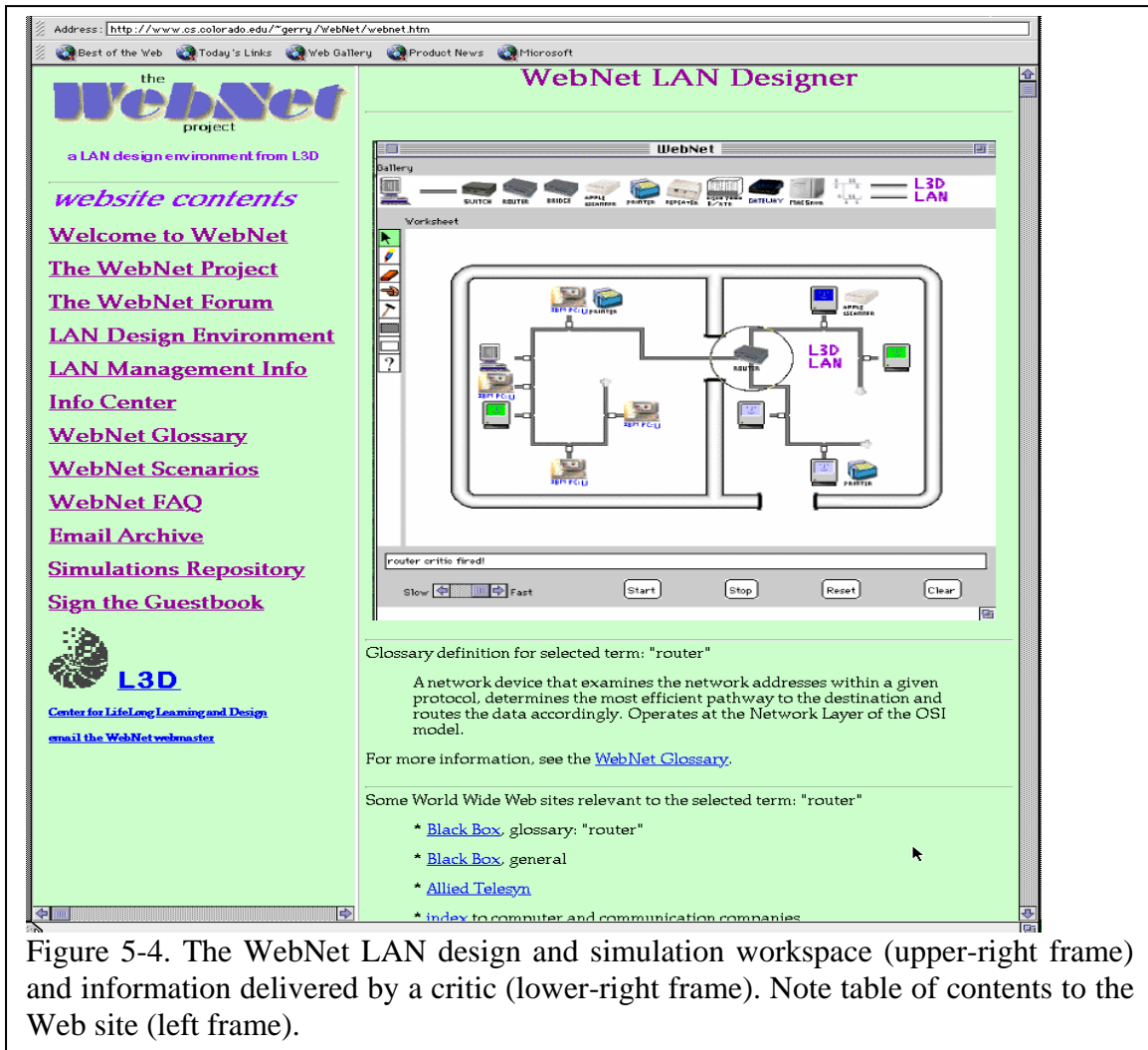


Figure 5-4. The WebNet LAN design and simulation workspace (upper-right frame) and information delivered by a critic (lower-right frame). Note table of contents to the Web site (left frame).

WebNet points to several email messages from Kay's colleagues that discuss router issues and how they have been handled locally. The Email Archive includes all emails sent to Kay's LAN management workgroup in the past. Relevant emails are retrieved and ordered by the Email Archive software (Lindstaedt, 1996) based on their semantic relatedness to a query. In Kay's situation, WebNet automatically generates a query describing the simulation context, particularly the need for a router. The repository can also be browsed, using a hierarchy of categories developed by the user community.

Kay reviews the email to find out which routers are preferred by her colleagues. Then she looks up the latest specs, options and costs on the Web pages of router suppliers. Kay adds the router she wants to the simulation and re-runs the simulation to check it. She saves her new design in a catalog of local LAN layouts. Then she sends an email message to her co-workers telling them to take a look at the new design in WebNet's catalog. She also asks Jay, her mentor at Network Services, to check her work.

3.3.2 Interactive and Evolving Knowledge

Jay studies Kay's design in his Web browser. He realizes that the Iris computer that Kay has added is powerful enough to perform the routing function itself. He knows that

this knowledge has to be added to the simulation in order to make this option obvious to novices like Kay when they work in the simulation. `Agentsheets` includes an end-user programming language that allows Jay to reprogram the Iris workstation agent (Repenning, 1994). To see how other people have programmed similar functionality, Jay finds a server agent in the `Simulations Repository` and looks at its program. He adapts it to modify the behavior of the Iris agent and stores this agent back in the repository. Then he redefines the router critic rule in the simulation. He also sends Kay an email describing the advantages of doing the routing in software on the Iris; `WebNet` may make this email available to people in situations like Kay's in the future.

When he is finished, Jay tests his changes by going through the process that Kay followed. This time, the definition of router supplied by `WebNet` catches his eye. He realizes that this definition could also include knowledge about the option of performing routing in workstation software. The definitions that `WebNet` provides are stored in an interactive glossary. Jay goes to the `WebNet` glossary entry for "router" and clicks on the "Edit Definition" button. He adds a sentence to the existing definition, noting that routing can sometimes be performed by server software. He saves this definition and then clicks on "Make Annotations." This lets him add a comment suggesting that readers look at the simulation he has just modified for an example of software routing. Other community members may add their own comments, expressing their views of the pros and cons of this approach. Any glossary user can quickly review the history of definitions and comments—as well as contribute their own thoughts.

3.3.3 Community Memory

It is now two years later. Kay has graduated and been replaced by Bea. The subnet that Kay had added crashed last night due to print queue problems. Bea uses the `LAN Management Information` component of `WebNet` to trace back through a series of email trouble reports and entries in `LAN diaries`. The `LAN Management Information` component of `WebNet` consists of four integrated information sources: a `Trouble Queue` of reported problems, a `Host Table` listing device configurations, a `LAN Diary` detailing chronological modifications to the LAN and a `Technical Glossary` defining local hardware names and aliases. These four sources are accessed through a common interface that provides for interactivity and linking of related items.

The particular problem that Bea is working on was submitted to her through the `Trouble Queue`. Bea starts her investigation with the `Host Table`, reviewing how the printer, routers and servers have been configured. This information includes links to `LAN Diary` entries dating back to Kay's work and providing the rationale for how decisions were made by the various people who managed the LAN. Bea also searches the `Trouble Queue` for incidents involving the print queue and related device configurations. Many of the relevant entries in the four sources are linked together, providing paths to guide Bea on an insightful path through the community history. After successfully debugging the problem using the community memory stored in `WebNet`, Bea documents the solution by making entries and new cross links in the `LAN Management Information` sources: the `Trouble Queue`, `Host Table`, `LAN Diary` and `Glossary`.

In this scenario, Kay, Jay and Bea have used WebNet as a design, communication and memory system to support both their immediate tasks and the future work of their community. Knowledge has been constructed by people working on their own, but within a community context. Their knowledge has been integrated within a multi-component community memory that provides support for further knowledge building. This scenario—in which simulations, various repositories, electronic diaries, communication media and other utilities are integrated with work processes—suggests how complexly integrated CIEs can support communities of practice.

4. Perspectives on Shared, Evolving Knowledge Construction

In this section I propose a mechanism designed to make a CIE, like WebNet, more effective in supporting the interactions between individuals and groups in communities of practice. I call this mechanism “perspectives.” The perspectives mechanism permits a shared repository of knowledge to be structured in ways that allow for both individual work and the negotiation of shared results. To illustrate this approach to collaboration, I describe a CIE called WebGuide, which is an example of computer-supported collaborative learning (CSCL) (Crook, 1994; Koschmann, 1996b; O’Malley, 1995). The approach of interpretive, computational perspectives was proposed in chapter 4; the description of WebGuide continues in chapter 6.

4.1 Perspectives: A Collaboration Support Mechanism

The concept of perspectives comes from the hermeneutic philosophy of interpretation of Heidegger (1927/1996) and Gadamer (1960/1988). According to this philosophy, all understanding is situated within interpretive perspectives: knowledge is fundamentally perspectival. This is in accord with recent work in cognitive science that argues for theories of socially situated activity (Lave & Wenger, 1991; Winograd & Flores, 1986). These theories extend the hermeneutic approach to take into account the role of social structures in contributing to molding the construction of knowledge (Vygotsky, 1930/1978). Communities of practice play an important role in the social construction of knowledge (Brown & Duguid, 1991).

Knowledge here is the interpretation of information as meaningful within the context of personal and/or group perspectives. Such interpretation by individuals is typically an automatic and tacit process of which people are not aware (see chapter 4). It is generally supported by cultural habits (Bourdieu, 1972/1995) and partakes of processes of social structuration (Giddens, 1984b). This tacit and subjective personal opinion evolves into shared knowledge primarily through communication and argumentation within groups (Habermas, 1981/1984).

Collaborative work typically involves both individual and group activities. Individuals engage in personal *perspective-making* and also collaborate in *perspective-taking* (Boland & Tenkasi, 1995). That is, individuals construct not only elements of domain knowledge, but also their own “take” on the domain, a way of understanding the network of knowledge that makes up the domain. An essential aspect of creating one’s perspective on a domain of knowledge is to take on the perspectives of other people in the community. Learning to interpret the world through someone else’s eyes and then

adopting this view as part of one's own intellectual repertoire is a fundamental mechanism of learning. Collaborative learning can be viewed as a dialectic between these two processes of perspective making and perspective taking. This interaction takes place at both the individual and group units of analysis—and it is a primary mode of interchange between the two levels.

While the Web provides an obvious medium for collaborative work, it provides no support for the interplay of individual and group understanding that drives collaboration. First, we need ways to find and work with information that matches our personal needs, interests and capabilities. Then we need means for bringing our individual knowledge together to build shared understanding and collaborative products. Enhancing the Web with perspectives may be an effective way to accomplish this.

As a mechanism for computer-based information systems, the term *perspective* means that a particular, restricted segment of an information repository is being considered, stored, categorized and annotated. This segment consists of the information that is relevant to a particular person or group, possibly personalized in its display or organization to the needs and interests of that individual or team. Computer support for perspectives allows people in a group to interact with a shared community memory; everyone views and maintains their own perspective on the information without interfering with content displayed in the perspectives of other group members.

One problem that typically arises is that isolated perspectives of group members tend to diverge instead of converge as work proceeds. Structuring perspectives to encourage perspective-taking, sharing and negotiation offers a solution to this by allowing members of a group to communicate about what information to include as mutually acceptable. The problem with negotiation is generally that it delays work on information while potentially lengthy negotiations are underway. Here, a careful structuring of perspectives provides a solution, allowing work to continue within personal perspectives while the contents of shared perspectives are being negotiated. I believe that perspectives structured for negotiation is an important approach that can provide powerful support for collaborative use of large information spaces on the Web.

The idea of computer-based perspectives traces its lineage to hypertext ideas like “trail blazing” (Bush, 1945), “transclusion” (Nelson, 1981) and “virtual copies” (Mittal, Bobrow, & Kahn, 1986)—techniques for defining and sharing alternative views on large hypermedia spaces. At the University of Colorado, we have been building desktop applications with perspectives for the past decade (see (McCall *et al.*, 1990) and chapters 1 and 4) and are now starting to use perspectives on the Web.

Earlier versions of the perspectives mechanism defined different contexts associated with items of information. For instance, in an architectural DODE, information about electrical systems could be grouped in an “electrical context” or “electrician's perspective.” In a CIE, this mechanism is used to support collaboration by defining personal and group perspectives in which collaborating individuals can develop their own ideas and negotiate shared positions. These informational contexts can come to represent perspectives on knowledge. While some collaboration support systems provide personal and/or group workspaces (Scardamalia & Bereiter, 1996), the perspectives implementation described below is innovative in supporting hierarchies or graphs of perspective inheritance.

This new model of perspectives has the important advantage of letting team members inherit the content of their team's perspective and other information sources without having to generate it from scratch. They can then experiment with this content on their own without worrying about affecting what others see. This is advantageous as long as one only wants to use someone else's information to develop one's own perspective. It has frequently been noted in computer science literature (Boland & Tenkasi, 1995; Floyd, 1992) that different stakeholders engaged in the development and use of a system (e.g., designers, testers, marketing, management, end-users) always think about and judge issues from different perspectives and that these differences must be taken into account.

However, if one wants to influence the content of team members' perspectives, then this approach is limited because one cannot change someone else's content directly. It is of course important for supporting collaborative work that the perspectives maintain at least a partial overlap of their contents in order to reach successful mutual understanding and coordination. The underlying subjective opinions must be intertwined to establish intersubjective understanding (Habermas, 1981/1984; Tomasello *et al.*, 1993). In the late 1990's, our research has explored how to support the intertwining of perspectives using the perspectives mechanism for CIEs.

4.2 Designing a System for Collaborative Knowledge Construction

We designed a system of computational support for interpretive perspectives in which content of one perspective can be automatically inherited into perspectives connected in a perspective hierarchy or graph. This sub-section recounts the motivation and history of the design of our integration of the perspectives mechanism into a CIE named WebGuide. It discusses a context in which student researchers in middle school learn how to engage in collaborative work and how to use computer technologies to support their work.

In summer 1997 we decided to apply our vision of intertwining personal and group perspectives to a situation in middle school (12-year-old 6th graders) classrooms. The immediate presenting problem was that students could not keep track of website addresses they found during their Web research. The larger issue was how to support team projects. We focused on a project-based curriculum (Blumenfeld *et al.*, 1991) on ancient civilizations of Latin America (Aztec, Inca, Maya) used at the school.

In compiling a list of requirements for WebGuide, we focused on how computer support can help structure the merging of individual ideas into group results. Such support should begin early and continue throughout the student research process. It should scaffold and facilitate the group decision-making process so that students can learn how to build consensus. WebGuide combines displays of individual work with the emerging group view. Note that the topic on Aztec Religion in figure 5-5 was added to the team perspective by another student (Bea). Also note that Kay has made a copy of a topic from Que's perspective so she can keep track of his work related to her topic. The third topic is an idea that Kay is preparing to work on herself. Within her personal electronic workspace, Kay inherits information from other perspectives (such as her team perspective) along with her own work.

Figure 5-5 goes approximately here

It soon became clear to us that each student should be able to view the notes of other team members as they work on common topics, not only after certain notes are accepted by the whole team and copied to the team perspective. Students should be able to adopt individual items from the work of other students into their own perspective, in order to

start
the

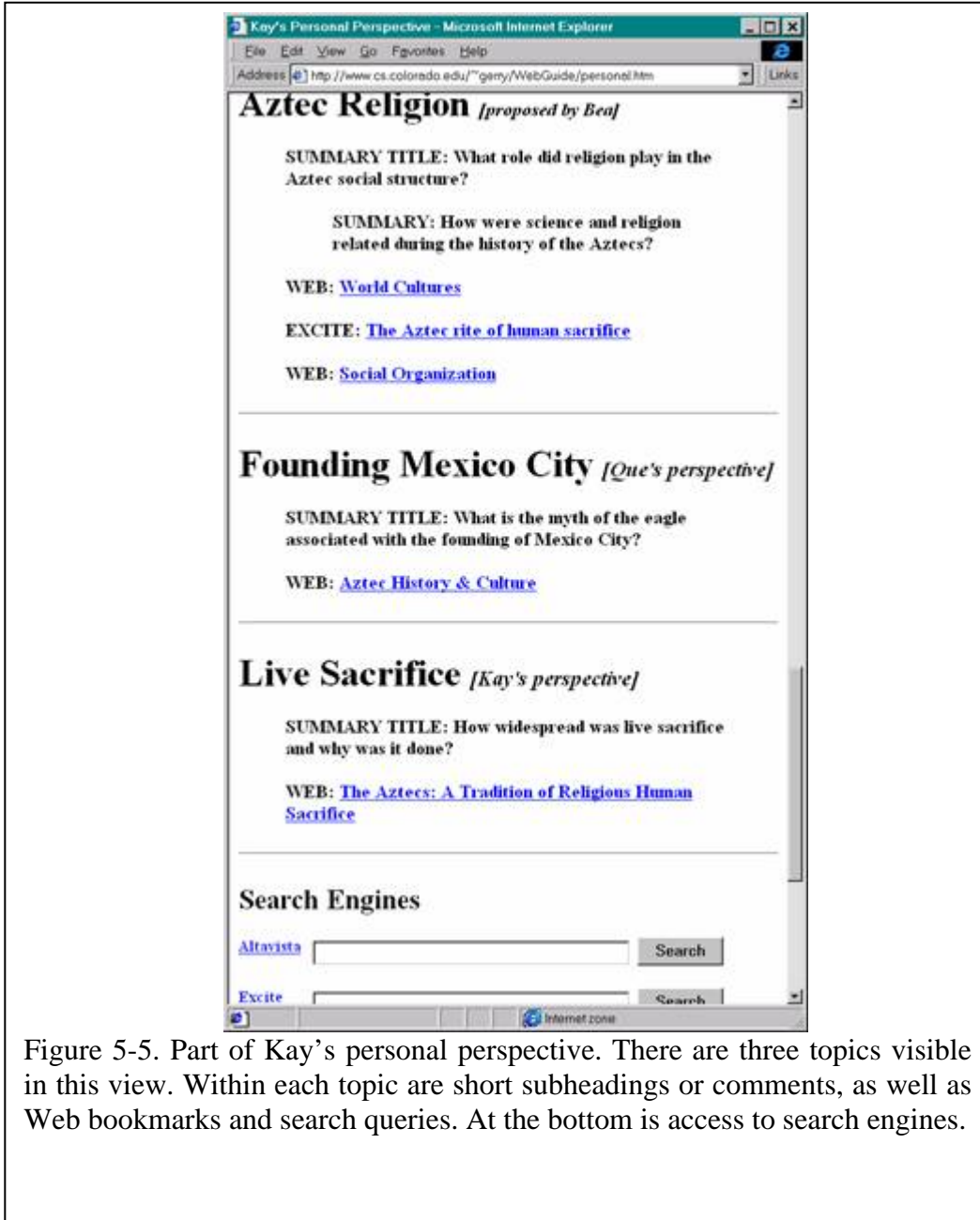


Figure 5-5. Part of Kay's personal perspective. There are three topics visible in this view. Within each topic are short subheadings or comments, as well as Web bookmarks and search queries. At the bottom is access to search engines.

collaboration and integration process. From early on, they should be able to make proposals for moving specific items from their personal perspective (or from the perspective of another) into the team perspective, which will eventually represent their team product, the integration of all their work.

The requirement that items of information can be copied, modified and rearranged presupposes that information can be collected and presented in small pieces—at the granularity of a paragraph or an idea. This is also necessary for negotiating which pieces should be accepted, modified, or deleted. We want the CIE to provide extensive support for collecting, revising, organizing and relating ideas as part of the collaborative construction of knowledge.

The Web pages of a student's personal perspective should not only contain live link bookmarks and search queries, but also categories, comments and summaries authored by the student. Comments can optionally be attached to any information item. Every item is tagged with the name of the person who created or last modified it. Items are also labeled with perspective information and time stamps.

Students each enter notes in their personal perspectives using information available to them: the Web, books, encyclopedia, CD-ROM, discussions, or other sources. Students can review the notes in the class perspective, their team perspective and the personal perspectives of their team mates. All of these contents are collected in comparison perspectives, where they are labeled by their perspective of origin. Students extract from the group research those items which are of interest to them. Then, within their personal perspectives they organize and develop the data they have collected by categorizing, summarizing, labeling and annotating. The stages of investigating, collecting and editing can be repeated as many times as desired. Team members then negotiate which notes should be promoted to the team perspective to represent their collaborative product.

The class project ends with each team producing an organized team perspective on one of the civilizations. These perspectives can be viewed by members of the other teams to learn about the civilizations that they did not personally research. The team perspectives can also provide a basis for additional class projects, like narrative reports and physical displays. Finally, this year's research products can be used to create next year's class perspective starting point, so new researchers can pick up where the previous generation left off—within a Web information space that will have evolved substantially in the meantime.

4.3 Supporting Perspective-Making

The application of a CIE to the problem of supporting middle school students conducting Web research on the Aztec, Maya and Inca civilizations drove the original concept of WebGuide. Since then, the basic functionality of the CIE has been implemented as a Java applet and applied in two other applications: (1) *Gamble Gulch*: a set of middle school teams constructing conflicting perspectives on a local environmental problem and (2) *Readings '99*: a university research group exploring cognitive science theories that have motivated the WebGuide approach. These two applications further illustrate how perspective-making and perspective-taking can be supported within a CIE. They are briefly discussed here, but will be described in more detail in chapter 6.

We first used an early implementation of WebGuide in a classroom at the Logan School for Creative Learning in Denver (see figure 5-6). For the previous five years, this class of middle school students had researched the environmental damage done to mountain streams by "acid mine drainage" from deserted gold mines in the Rocky Mountains above Denver. They actually solved the problem at the source of a stream coming into Boulder from the Gamble Gulch mine site by building a wetlands area to

filter out heavy metals. Now they were investigating the broader ramifications of their past successes; they were looking at the issue of acid mine drainage from various alternative—and presumably conflicting—perspectives. The students interview adult mentors to get opinions from specific perspectives: environmental, governmental, mine-owner and local landowners.

Figure 5-6 goes approximately here

As an initial field test of the WebGuide system, this trial resulted in valuable experience in the practicalities of deploying such a sophisticated program to young students over the Web. The students were enthusiastic users of the system and offered (through WebGuide) many ideas for improvements to the interface and the functionality. Consequently, WebGuide benefited from rapid cycles of participatory design. The differing viewpoints, expectations and realities of the software developers, teachers and students provided a dynamic field of constraints and tensions within which

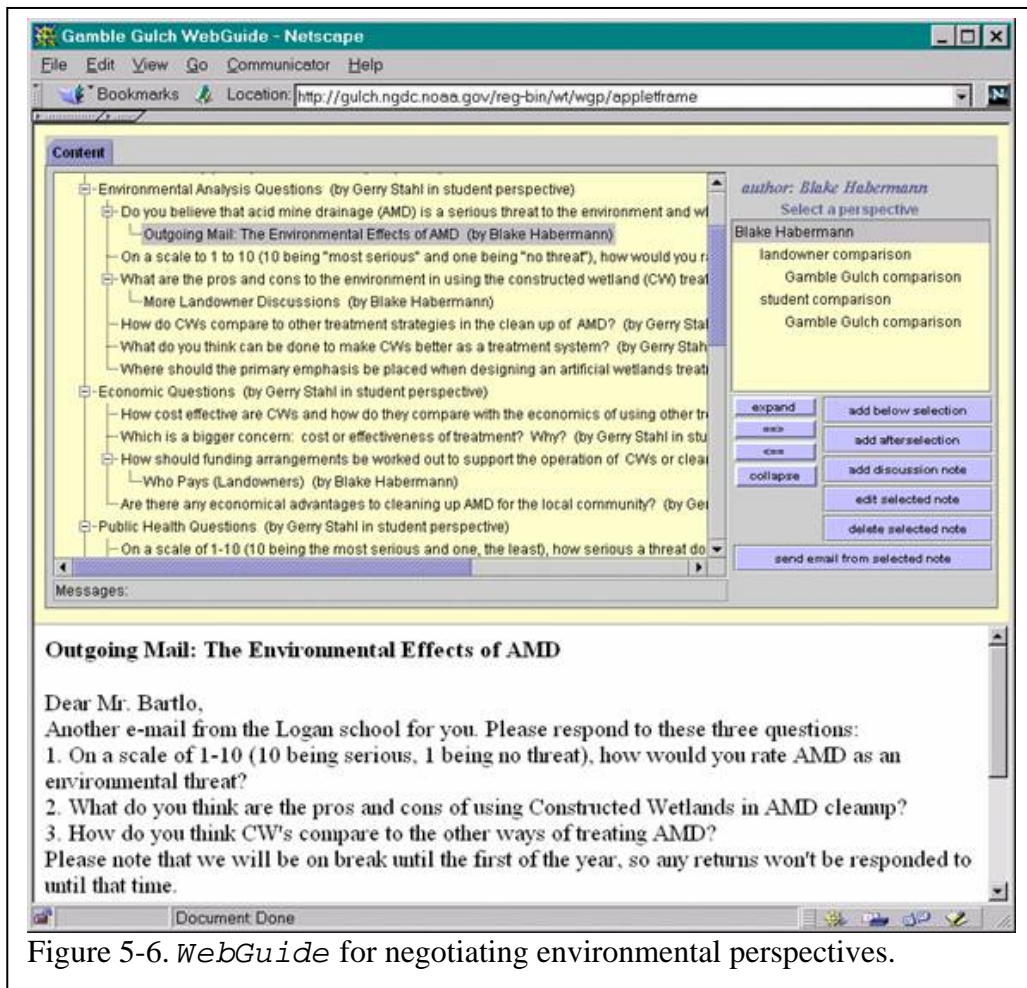


Figure 5-6. *WebGuide* for negotiating environmental perspectives.

the software, its goals and the understanding of the different participants co-evolved within a complex structural coupling.

The Readings '99 application of WebGuide the following year stressed the use of perspectives for structuring collaborative efforts to build shared knowledge. The goal of the graduate

seminar was to evolve sophisticated theoretical views on computer mediation within a medium that supports the sharing of tentative positions and documents the development of ideas and collaboration over time. A major hypothesis to be explored by the course was that software environments with perspectives—like WebGuide—can provide powerful tools for coordinated intellectual work and collaborative learning. For instance, it explored how the use of a shared persistent knowledge construction space can support more complex discussions than ephemeral face-to-face conversations.

This is not the place to evaluate the effectiveness of the WebGuide perspective mechanism. The story of its development will be continued in chapter 6. Here, I wanted simply to suggest the possibility of computational support for collaboration that goes beyond what is now commercially available. The perspectives mechanism allows people to work collaboratively by intertwining their personal and group perspectives on shared ideas.

5. Extending Human Cognition

Our early work on domain-oriented design environments (DODEs)—reviewed in section 2 of this chapter—was an effort to augment human intelligence within the context of professional design activities. At a practical level, our focus on building systems for experts (rather than expert systems) contrasted with much research at the time that emphasized either (1) artificial intelligence heuristics intended to automate design tasks or (2) user-friendly, idiot-proof, walk-up-and-use systems that were oriented toward novices. In theoretical terms, we acted upon the view that human intelligence is not some biologically fixed system that can be modeled by and possibly even replaced by computationally analogous software systems. Rather, human intelligence is an open-ended involvement in the world that is fundamentally shaped by the use of tools (Donald, 1991; Heidegger, 1927/1996; Vygotsky, 1930/1978). In this view, computer-based systems can extend the power of human cognition. Like any effective tools, software systems like DODEs mediate the cognitive tasks, transforming both the task and the cognitive process (Norman, 1993; Winograd & Flores, 1986). In addition, computer-based systems enhance the capabilities of their users by encapsulating the derived human intentionality of their developers (Stahl, 1993). In this light, we saw the emergence of the Web as offering an enabling technology for allowing communities of DODE users to embed their own collective experience in the critics and design rationale components of DODE knowledge bases.

The movement in our work from DODEs to collaborative information environments (CIEs)—reviewed in section 3—was not only driven by the potential of Web technology. It was also motivated by the increasing awareness of the socially situated character of contemporary work, including the important role of communities of practice (Brown & Duguid, 1991; Lave & Wenger, 1991; Orr, 1990). The fact that much work and learning is overtly collaborative these days is not accidental (Marx, 1867/1976). Just as the cognitive processes that are engaged in work and learning are fundamentally mediated by the tools that we use to acquire, store and communicate knowledge, they are equally mediated by social phenomena (Giddens, 1984b; Habermas, 1981/1984). In fact, tools, too, have a social origin, so that the mediation of human cognition results from complex interactions between the artifactual and the social (Orlikowski *et al.*, 1995; Vygotsky,

1930/1978). CIEs are designed to serve as socially-imbued, computationally powerful tools. They make the social character of knowledge explicit and they support collaborative knowledge building.

The notion of a perspectives mechanism such as the one prototyped in WebGuide—is reviewed in section 4—is to provide tool affordances that support the social nature of mediated cognition. Collaborative work and learning involve activities at two units of analysis: the individual and the group (Boland & Tenkasi, 1995; Orlikowski, 1992). Personal perspectives and team perspectives provide a structure for distinguishing these levels and create workspaces in which the different activities can take place. Of course, the crux of the problem is to facilitate interaction between these levels: the perspectives mechanism lets individuals and teams copy notes from one space to another, reorganize the ideas and modify the content. Communities of practice are not simple, fixed structures, and so the graph of perspective inheritance must be capable of being interactively extended to include new alliances and additional levels of intermediate sub-teams.

The perspectives mechanism (more fully discussed in chapter 6) has not been proposed as a complete solution; it is meant to be merely suggestive of computationally intensive facilities to aid collaboration. Systematic support for negotiating consensus building and for the promotion of agreed upon ideas up the hierarchy of sub-teams is an obvious next step (see chapters 7 & 8). Collaborative intelligence places a heavy cognitive load on participants; any help from the computer in tracking ideas and their status would free human minds for the tasks that require interpretation of meaning (see chapter 16).

The concept of intelligence underlying the work discussed in this chapter views human cognition, software processing and social contexts as complexly and inseparably intertwined. In today's workplaces and learning milieus, neither human nor machine intelligence exists independently of the other. Social concerns about AI artifacts are not secondary worries that arise after the fact, but symptoms of the fundamentally social character of all artifacts and of all processes of material production and knowledge creation (Marx, 1867/1976; Vygotsky, 1930/1978). I am trying to explore the positive implications of this view by designing collaborative information environments to support knowledge construction by small groups within communities.