

Collaborative Information Environments to Support Knowledge Construction by Communities

Gerry Stahl

Center for LifeLong Learning and Design and Institute of Cognitive Science, University of Colorado at Boulder, CO, USA

Abstract: 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 paper reviews three stages of work on artificial (computer-based and Web-based) systems that augment the intelligence of people and organisations. 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.

Keywords: Collaborative learning; Communities of practice; Construction of knowledge; Perspectives mechanism

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

The creation of innovative artefacts and helpful knowledge in our complex world – with its refined division of labour 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 and 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, 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 and Schön, 1978).
- Learning – especially collaborative learning – has become a new form of labour, an integral component of work and organisations (Lave and Wenger, 1991).
- Individual memory, attention and understanding are too limited for today's complex tasks; divisions of labour are constantly shifting and learning is required to coordinate and respond to the changing demands on community members (Brown and Duguid, 1991).
- Learning necessarily includes organisational 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, organise and share knowledge by supporting these processes. However, the design of such software remains an open research area (Stahl, 1998).

The contemporary need to extend the learning process from schooling into organisational and community realms is known as *lifelong learning*. Our past research at the University of Colorado's Centre for LifeLong Learning and Design explored the computer support of lifelong learning with what we call *domain-oriented design environments* (DODEs). This paper 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. The paper illustrates three stages our efforts have gone through in this direction during the current decade with illustrative software systems.

Section 2 illustrates 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

organisational learning in collaborative work settings known as communities of practice; a scenario of a prototype system called WebNet illustrates this. Section 4 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 applications. A concluding section locates this discussion within the context of AI and society.

2. Augmenting the Work of Individual Designers

In this section we 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 et al., 1997) and software development (Robbins and 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 recognise 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 into the system knowledge of the domain, including design objects and design rationale. A DODE typically provides a computational workspace within which a designer can construct an artefact and represent components of the artefact being constructed. Unlike a computer-aided design (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 paper) allows a designer to construct a network design by arranging items from a palette representing workstations, servers, routers, cables and other devices from the LAN domain. 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 used a mechanism we call *critiquing* (Fischer et al., 1993/1998). 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 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 learn to use its rich set of capabilities (see Fig. 1). NetSuite contains a wealth of domain knowledge. Its palette of devices that can be placed in the construction area numbers over 5000, with more downloadable 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.

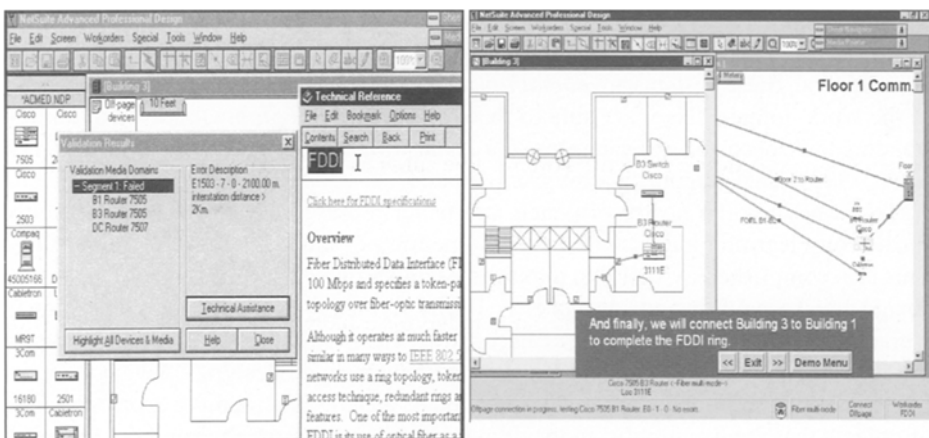


Fig. 1. Two views of NetSuite. In the left view, the system has noted that a cable length specification for an FDDI network has been exceeded in the design and the system has delivered information about the specification and affected devices. In the right view, parts of the network viewed in physical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

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 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

Based on our understanding of organisational 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 organisations rely on home-grown information systems because they believe that critical parts of their local information are unique. A community of practice has its own ways of doing things. Generally, these local practices are understood tacitly and are propagated through apprenticeship (Lave and 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 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 minimise the effort required. This means:

1. The DODE knowledge base should be integrated with work practices in ways that capture knowledge as it is created.
2. The benefits of maintaining the knowledge base have to be clearly experienced by participants.
3. There may need to be an accepted distribution of roles related to the functioning of the organisational memory.
4. The software environment must be thoroughly interactive so that users can easily enter data and comments.
5. 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.

6. As the information space grows, there should be ways for people to restructure it so that its organisation and functionality keep pace with its evolving contents and uses (Fischer et al., 1997).

DODEs must be extended in these ways to support communities of practice, 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, we 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 labour 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 socialisation, apprenticeship, training, schooling and culture (Bourdieu, 1972; Giddens, 1984; Lave and Wenger, 1991), as well as by explicit standards. Often, work is performed by collaborating teams that form *communities of practice* within or across organisations (Brown & Duguid, 1991). These communities evolve their own styles of communication and expression, or genres (Bakhtin, 1986; Yates and Orlikowski, 1992).

For instance, interviews we conducted showed that computer network managers 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 organisation (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 organisational memory technologies by specific communities involves complex

social issues beyond the scope of this paper. For a review of common 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 realised. External memories, incorporating symbolic representations, facilitated the growth of complex societies and sophisticated scientific understanding. Their effectiveness relied upon the spread of literacy and industrialisation. 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 organisational memories must be matched with new social structures that produce and reproduce patterns of organisational learning (Giddens, 1984; Lave and Wenger, 1991).

Community memories are to communities of practice what human memories are to individuals. They make use of explicit, external, symbolic representations that allow for shared understanding within a community. They make organisational learning possible within the group (Ackerman and McDonald, 1996; Argyris and Schön, 1978; Borghoff and Parechi, 1998; Buckingham Shum and Hammond, 1994; Senge, 1990).

3.1.3 Integrative Systems for Community Memory

Effective community memory relies on integration. Tools for representing design artefacts and other work tasks must be related to rich repositories of information that can be brought to bear when needed. Communication about artefacts under development should be tied to the artefact 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 interrelated 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 interrelationships. For instance, in a system like those to be discussed in section 4 using perspectives, 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 worldwide. 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 to or want to, and can communicate about it. This 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 tacit existing 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

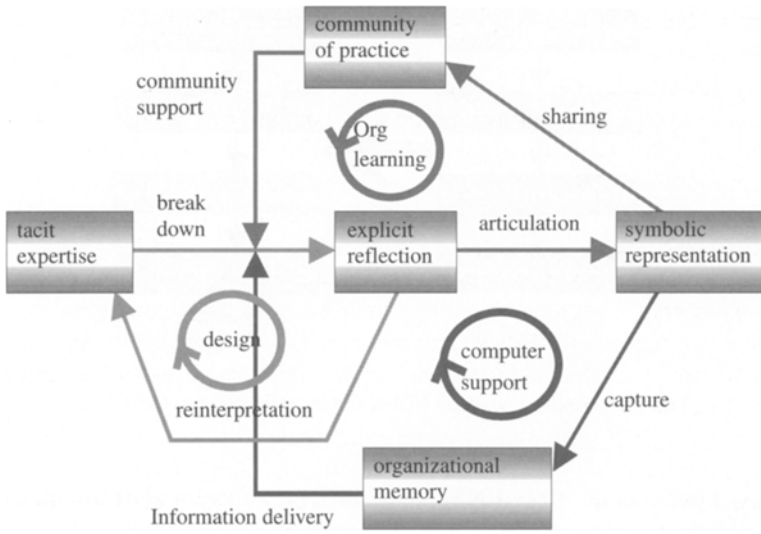


Fig. 2. Cycles of design, computer support, and organisational learning.

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 Fig. 2, cycle on left) (Stahl, 1993a).

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 *organisational learning* (see Fig. 2, upper cycle). The personal reflection and collaborative articulation of shared perspectives make innovation possible (Boland and Tenkasi, 1995; Tomasello et al., 1993).

Organisational learning can be supported by computer-based systems of organisational memory if the articulated knowledge is captured in a digital symbolic representation. The information must be stored and organised in a format that facilitates its subsequent identification and retrieval. In order to provide *computer support*, the software must be able to recognise breakdown situations when particular items of stored information might be useful to human reflection (see Fig. 2, lower cycle) (Stahl, 1993b). DODEs provide computer support for design by individuals. They need to be extended to collaborative information environments (CIEs) to support organisational 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

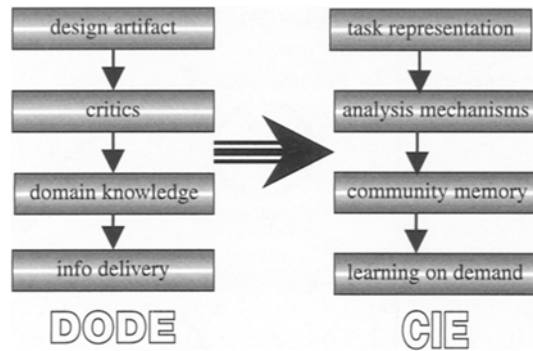


Fig. 3. Generalisation of the DODE architecture (left) to a CIE (right).

right way (Fischer et al., 1993/1998). To do this, the software must be able to analyse 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 artefact* being constructed in the design environment work space (see Fig. 3, left).

One can generalise from the critiquing approach of these DODEs to arrive at an overall architecture for organisational 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 generalise 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 artefact 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 a rich information space there may be many relationships of interest between new work artefacts and items in the organisational 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 from the *task representation* to relevant *community memory* information items to support *learning on demand* (see Fig. 3, right).

To take a very different example, suppose you are writing a paper within a software environment that includes a digital library of papers written by you and your colleagues. Then an analysis mechanism to support your learning might compare sentences or

paragraphs in your 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 deliver for your learning. We use latent semantic analysis (Landauer and Dumais, 1997) to mine our email repository (Lindstaedt and Schneider, 1997) and are exploring similar uses of this mechanism to link task representations to textual information to support organisational learning. Other retrieval mechanisms might be appropriate for mining catalogues of software agents or components, design elements and other sorts of organisational memories.

Using our example of LAN design, we next show how a CIE might function in this domain. We present a scenario of use of WebNet, a prototype we 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 Fig. 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 behaviour of an Iris workstation. She adds this to her palette and to her design.

When Kay runs the LAN simulation, WebNet proactively inserts a router (see Fig. 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 Fig. 4, lower right). Here, WebNet has signalled 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.

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 reruns the simulation

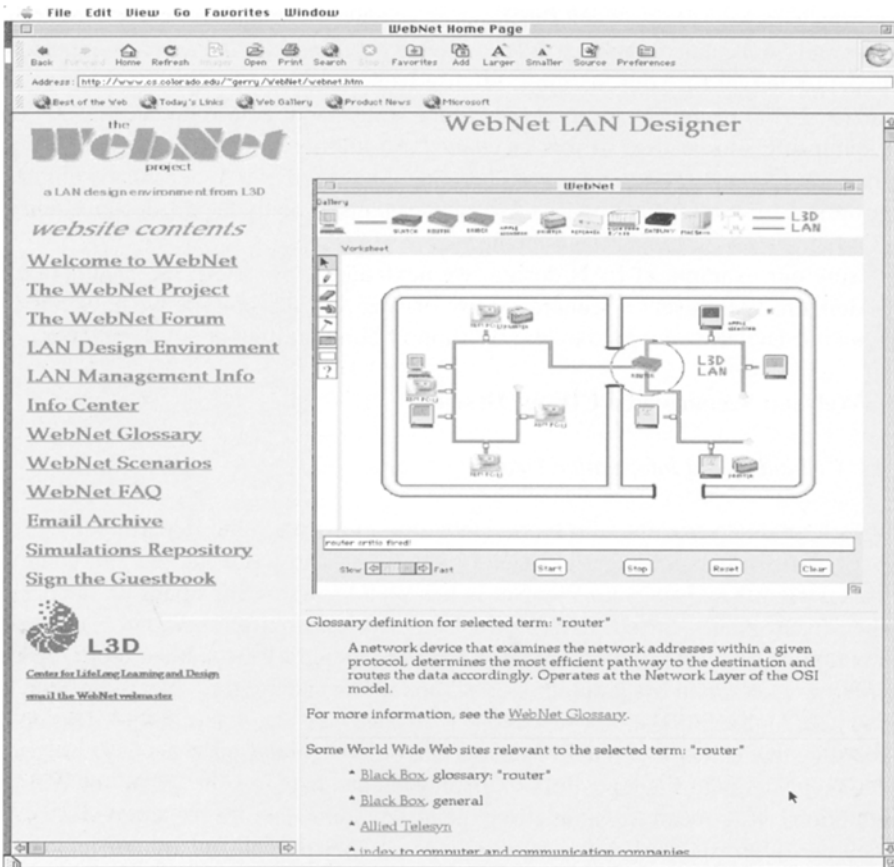


Fig. 4. The WebNet LAN design and simulation workspace (upper-right frame) and information delivered by a critic (lower-right frame). Note table of contents to the Web site (left frame).

to check it. She saves her new design in a catalogue 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 catalogue. 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 realises 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 behaviour 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 realises 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 Info 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 we propose a mechanism designed to make a CIE like WebNet more effective in supporting the interactions between individuals and groups in communities of practice. We call this mechanism ‘perspectives’. The perspectives mechanism allows a shared repository of knowledge to be structured in ways that allow for both individual work and the negotiation of shared results. In the terms of the analysis presented in Fig. 2, NetSuite supports individual design, WebNet supports organisational memory and WebGuide supports organisational learning. To illustrate this approach to collaboration, we describe a CIE called WebGuide, which is an example of computer-supported collaborative learning (CSCL) (Crook, 1994; Koschmann, 1996; O’Malley, 1995).

4.1 Perspectives: A Collaboration Support Mechanism

The concept of perspectives comes from the hermeneutic philosophy of interpretation of Heidegger and Gadamer (Gadamer, 1960/1988; Heidegger, 1927/1996). 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 and Wenger, 1991; Winograd and Flores, 1986). These theories extend the hermeneutic approach to take into account the role of social structures in contributing to moulding the construction of knowledge (Vygotsky, 1930/1978). Communities of practice play an important role in the social construction of knowledge (Brown and 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 that people are not aware of (Polanyi, 1962; Stahl, 1993b). It is generally supported by cultural habits (Bourdieu, 1972) and partakes of processes of social structuration (Giddens, 1984). This tacit and subjective personal opinion evolves into shared knowledge primarily through communication and argumentation within groups (Habermas, 1981).

Collaborative work typically involves both individual and group activities. Individuals engage in personal *perspective making* and also collaborate in *perspective taking* (Boland and 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 making 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 levels 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 match our personal needs, interests and capabilities. Then we need means for bringing our individual knowledge together to build a 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, categorised and annotated. This segment consists of the information that is relevant to a particular person or group, possibly personalised in its display or organisation to the needs and interests of that individual or team (Stahl, 1995). 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 converging 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 under way. 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. We believe that perspectives structured for negotiation constitute an important approach that can provide powerful support for collaborative use of large information spaces on the Web.

The idea of perspectives traces its lineage to hypertext ideas like 'trail blazing' (Bush, 1945), 'transclusion' (Nelson, 1981) and 'virtual copies' (Mittal et al., 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 (McCall et al., 1990; Stahl, 1995; Stahl et al., 1995) 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 work spaces (e.g., (Scardamalia & Bereiter, 1996)), the perspectives implementation described below is innovative in supporting hierarchies of perspective inheritance.

The most important characteristics of the perspective mechanism (Stahl, 1993a) that we have been exploring are:

- Individual community members have access to what appears to be their own information source. This is called their *personal perspective*. It consists of items

from a shared central information repository that are tagged as being visible within that particular perspective (or in any perspective inherited by that perspective).

- Community member A can integrate an item from B's perspective into A's personal perspective by creating a link or *virtual copy* of the item. If B modifies the original item, then it changes in A's perspective as well. However, if A modifies the item, a new item is actually created for A, so that B's perspective is not changed. This arrangement generally makes sense because A wants to view (or inherit) B's item, even if it evolves. However, B should not be affected by the actions of someone who copied one of B's items.
- Alternatively, A can *physically copy* the contents of an item from B's perspective. In this case, the copies are not linked to each other in any way. Since A and B are viewing physically distinct items now, either can make changes without affecting the other's perspective.
- When A creates a virtual copy of an item from B's perspective, A can decide if she will also get virtual copies of items related to that one, or if she will create her own sub-network for her copy of that item. Arbitrarily large sub-networks of information can be inherited with no overhead using the virtual copy mechanism.
- Items of information can be created, edited, rearranged, linked together or deleted by users within their personal perspective without affecting the work of others.
- New perspectives can be created by users. Perspectives can inherit from existing perspectives. Thus, a team perspective can be created that includes virtual copies of all contents of the inherited perspectives of the team members.
- There is an inheritance tree of perspectives; descendants inherit the contents of their ancestor perspectives. Changes (additions, edits, deletions) in the ancestor are seen in descendent perspectives, but not vice versa.
- A hierarchy of team, sub-team and individual perspectives can be built to match the needs of a particular community.

This 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 and 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; Tomasello et al., 1993). In the past two years, 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

This subsection 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 future researchers in middle school learn how to engage in collaborative work and how to use computer technologies to support their work.

4.2.1 Supporting Collaborative Student Web Research

In summer 1997 we decided to apply our vision of intertwining personal and group perspectives to a situation in middle school (6th grade, 12-year-olds) classrooms we work with. The immediate presenting problem was that students could not keep

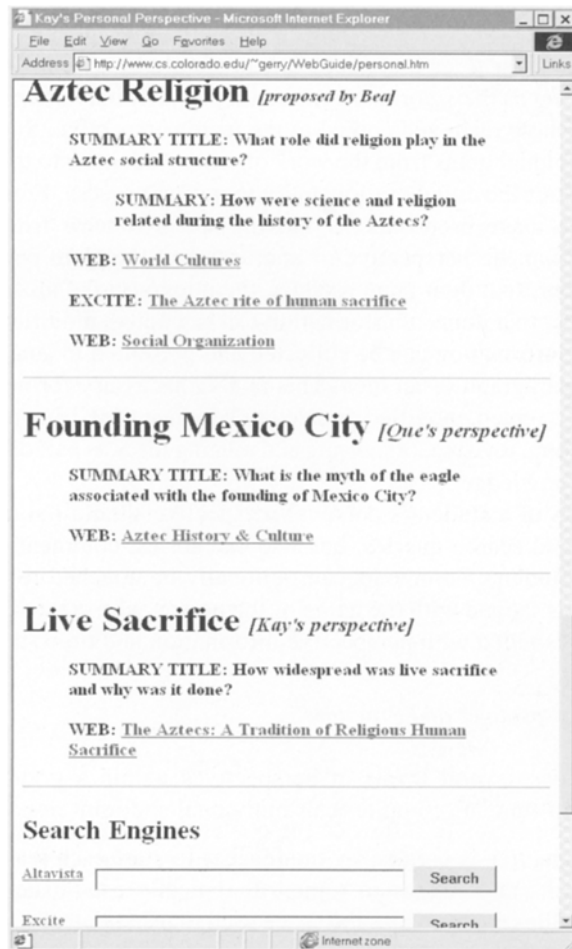


Fig. 5. Part of Kay's personal perspective. There are three topics visible in this view. Within each topic are short subheadings or comments, as well as Web bookmarks and search queries. At the bottom is access to search engines.

track of Web site URLs 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 civilisations 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 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 Fig. 5 was added to the team perspective by another student (Bea). Also note that Kay has made a virtual 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 electronic workspace, Kay inherits information from other perspectives along with her own work.

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 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, organising 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 labelled with perspective information and time stamps.

4.2.2 *Types of Perspectives and Practices*

WebGuide provides several levels of perspectives within a graph of perspective inheritance to help students compile their individual and joint research:

- The *class perspective* is created by the teacher to start each team off with some initial bookmarks and suggested topics. It typically establishes a structure for classroom activities and provides the space used to instantiate the goal of collecting the products of collaborative intellectual work.
- The *team perspective* contains items that have been accepted by a team (like Bea's Aztec religion topic in Fig. 5). This perspective is pivotal; it gradually collects the products of the team effort.

- The student's *personal perspective* is a private work space. It inherits a view of everything in the team perspective. Thus, it displays the owner's own work within the context of items proposed or negotiated by the team and class – as modified by the student. Students can each modify (add, edit, delete, rearrange, link) their virtual copies of team items in their personal perspectives. They can also create completely new material there.
- The *comparison perspective* combines all the personal perspectives of team members and the team perspective, so that anyone can compare all the work that is going on. It inherits from the personal, team and class perspectives. Students can go here to get ideas and copy items into their own personal perspective or propose items for the team perspective.

To design software for collaborative learning in schools means to design curriculum and classroom process as well. Computer support has to be matched with appropriate content on the Web and with constructivist practices for knowledge-building communities (Scardamalia and Bereiter, 1991). The design of the WebGuide interface and the perspectives mechanism is accompanied by the design of informative Web pages and of a use scenario.

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 labelled by their perspective of origin. Students extract from the research those items which are of interest to them. Then, within their personal perspectives they organise and develop the data they have collected by categorising, summarising, labelling 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 organised group perspective on one of the civilisations. These perspectives can be viewed by members of the other teams to learn about the civilisations 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 WebGuide: 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 civilisations 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 teams constructing conflicting perspectives on a local environmental problem; and (2) *Readings '99*: a research group exploring cognitive science theories that have motivated the WebGuide approach. The following descriptions of these two

applications further illustrate how perspective making and perspective taking can be supported within a CIE.

4.3.1 Negotiating Environmental Perspectives

We are now using an early implementation of WebGuide in a classroom at the Logan School for Creative Learning in Denver (see Fig. 6). For the past five years, this class of middle school students has 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. This year they are investigating the broader ramifications of their past successes; they are 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.

The Gamble Gulch application of WebGuide serves as the medium through which the students collaboratively research these issues with their mentors and with each

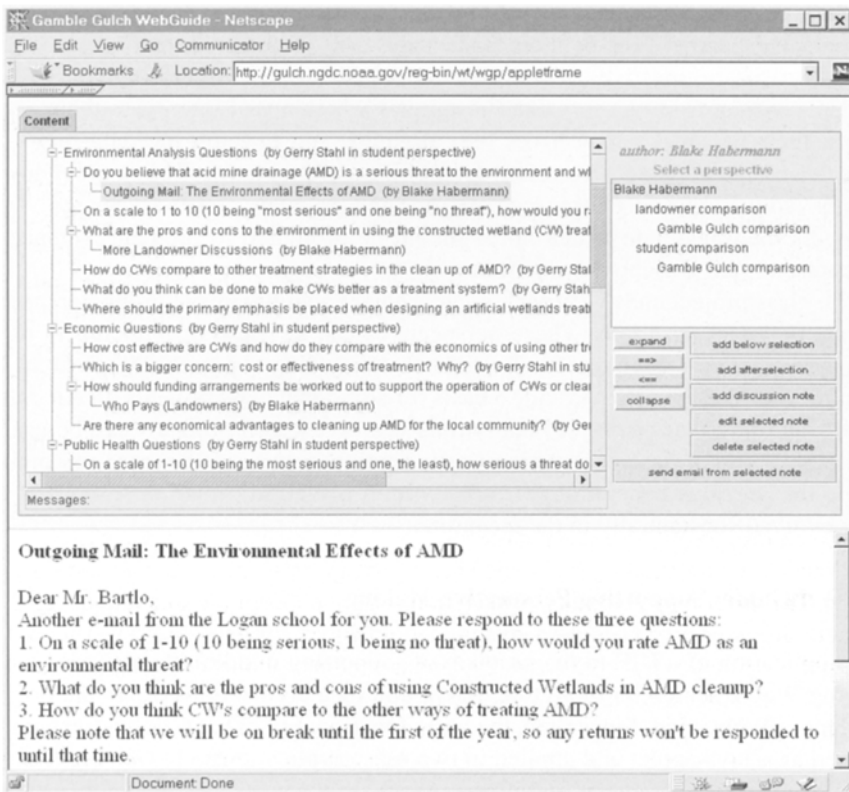


Fig. 6. WebGuide for negotiating environmental perspectives.

other. Each student and mentor has their personal perspective, and these perspectives inherit from one of the content-based team perspectives (environmental protection, governmental regulation, etc.), depending upon which intellectual perspective they are working on constructing. Even email interactions happen through WebGuide and are retained as notes in its perspectives.

Figure 6 shows one student's (Blake) personal perspective on the class discourse. The tree of discussion threads was 'seeded' with question categories, such as 'Environmental Analysis Questions'. Within these categories, the teacher posted specific questions for the students to explore, like 'Do you believe that acid mine drainage (AMD) is a serious threat to the environment?' Here, Blake has sent an email to one or more mentors asking for information related to this question. When replies are sent back, they will be automatically posted to the discussion tree under the original email. When someone clicks on a title in the tree, the contents of that item are displayed in an HTML frame below the applet (as is the body of the student's email in Fig. 6).

Blake is working in his personal perspective, which inherits from the class, student team and landowner team perspectives. He can add, edit and delete ideas in his perspective, as well as sending email in it. Because he is a member of the landowner team and the student group as well as the class, he can browse ideas in the student team comparison, the landowner team comparison and the Gamble Gulch class comparison perspective.

For this application, the teacher has decided that negotiation and perspective taking will take place in live classroom discussions, rather than in WebGuide. After a team or the whole class reaches a consensus, the teacher will enter the statements that they have agreed to into the team or class perspective.

The goal of the year-long course is not only to negotiate within teams to construct the various positions, but also to negotiate among the positions to reach consensus or to clarify differences. The teacher designed this class – with its use of WebGuide – to teach students that knowledge is perspectival, that different people construct views, compilations of facts and arguments differently depending upon their social situation. He hopes that his students will not only learn to evaluate statements as deriving from different perspectives, but also learn to negotiate the intertwining of perspectives to the extent that this is possible.

As an initial field testing of the WebGuide system, this trial has resulted in valuable experience in the practicalities of deploying such a sophisticated program to young students over the Web. The students are enthusiastic users of the system and offer (through WebGuide) many ideas for improvements to the interface and the functionality. Consequently, WebGuide is benefiting from rapid cycles of participatory design. The differing viewpoints, expectations and realities of the software developers, teachers and students provide a dynamic field of constraints and tensions within which the software, its goals and the understanding of the different participants co-evolve within a complex structural coupling.

4.3.2 Constructing Perspectives on Computer Mediation

We have recently begun an interdisciplinary graduate seminar on computer mediation of collaborative learning. The seminar uses WebGuide in several ways:

- *As the primary communication medium for their internal collaboration.* The seminar takes place largely on-line. Limited class time is used for people to get to know each other, to motivate the readings, to introduce themes that will be followed up on-line and to discuss how to use WebGuide within the seminar.
- *As an example CSCW system to analyse.* Highly theoretical readings on mediation and collaboration are made more concrete by discussing them in terms of what they mean in a system like WebGuide. The advantage of using a locally developed prototype like WebGuide as our example is that we not only know how it works in detail, but we can modify its functionality or appearance to try out suggestions that arise in the seminar.
- *As an electronic workspace for members to construct their individual and shared ideas.* Ideas entered into WebGuide persist there, where they can be revisited and annotated at any time. Ideas that arise early in the seminar will still be available in full detail later so that they can be related to new readings and insights. The record of discussions over a semester or a year will document how perspectives developed and interacted.

The Readings version of WebGuide incorporates a built-in permissions system that structures the social practices surrounding the use of the system. Seminar participants each have a home personal perspective in which they can manipulate notes however they like without affecting the views in other perspectives. They can add quick discussion notes or other kinds of statements. They can edit or delete anything within their home perspective. They can also make multiple copies or links (virtual copies) from notes in their personal perspective to other notes there. Anyone is free to browse in any perspective. However, if one is not in one's own perspective than one cannot

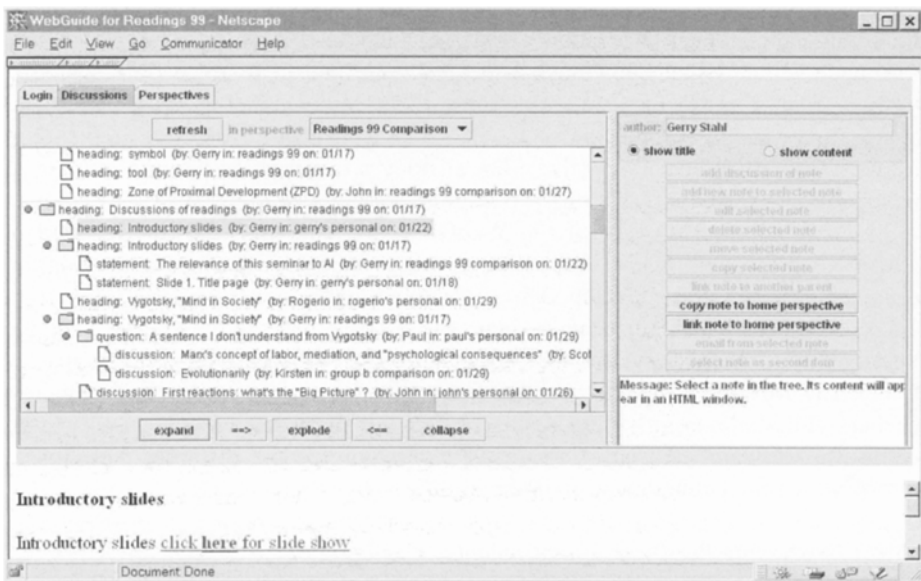


Fig. 7. WebGuide for constructing knowledge based on readings.

add, edit or delete notes there (as in Fig. 7). To manipulate notes freely, one must first copy or link the note into one's own personal perspective. The copy or link can optionally include copying (or virtual copying) all the notes below the selected note in the tree as well. These rules are enforced by the user interface, which checks whether or not someone is in their personal perspective and only allows the legal actions.

Students in the class can form subgroups either within or across their different disciplines. They develop ideas in their personal perspectives. They debate the ideas of other people by finding notes of interest in the Readings 99 Comparison perspective (or in a subgroup comparison perspective) and copying these notes into their own personal perspective, where they can comment on them. The clash of perspectives is visible in the comparison perspectives, while the personal perspectives allow for complete expression and organisation of a single perspective. This supports the taking of perspectives and the use of shared ideas in the making of perspectives.

The fact that an individual note may have different edited versions and different linking structures in different perspectives, that notes may have multiple parents within the discussion threads, that new perspectives can be added dynamically and may inherit from multiple other perspectives, sets WebGuide apart from simple threaded discussion media. It also makes the computations for displaying notes rather complex. This is a task that definitely requires computers. By relieving people of the equivalent of these display computations, computer support may allow people to collaborate more fluidly.

The Readings application of WebGuide stresses the use of perspectives for structuring collaborative efforts to build shared knowledge. The goal of the seminar is 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 is that software environments with perspectives – like WebGuide – can provide powerful tools for coordinated intellectual work and collaborative learning. For instance, it will explore how the use of a shared persistent knowledge construction space can support more complex discussions than ephemeral face-to-face conversations. We will explore the effectiveness of the Readings version of WebGuide as a computationally active tool to augment the knowledge construction work of a community (Stahl, 1998).

5. Conclusions: Extending Human Cognition

Our early work on domain-oriented design environments (DODEs) – reviewed in section 2 – 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 emphasised 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 modelled 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, 1993a). 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 is also motivated by the increasing awareness of the socially situated character of contemporary work, including the important role of communities of practice (Brown and Duguid, 1991; Lave and Wenger, 1991; Orr, 1990). The fact that much work and learning is overtly collaborative these days is not an accidental characteristic (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, 1984; Habermas, 1981). In fact, tools too have a social origin, so that the mediation of human cognition results from complex interactions between the artefactual and the social (Orlikowski et al., 1995; Vygotsky, 1930/1978). CIEs are designed to serve as socially embodied, 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 – 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 levels 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 (or virtually copy) notes from one space to another, reorganise the ideas and modify the content. Communities of practice are not simple structures, and so the graph of perspective inheritance can be interactively extended to include new alliances and additional levels of intermediate sub-teams.

The perspectives mechanism 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. 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 (Stahl, 1993a).

The concept of intelligence underlying the work discussed in this paper 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 artefacts are not secondary worries that arise after the fact, but symptoms of the fundamentally

social character of all artefacts and of all processes of material production and knowledge creation (Marx, 1867/1976; Vygotsky, 1930/1978). We are trying to explore the positive implications of this view by designing collaborative information environments to support knowledge construction by communities.

Acknowledgements

The earlier research on CIEs for design is a collaboration of the author with Gerhard Fischer and Jonathan Ostwald. We would like to thank the other members of the Center for LifeLong Learning and Design, particularly the 'Organizational Memory and Organizational Learning' group, including Jay Smith, Scott Berkebile, Sam Stoller, Jim Masson, and Tim Ohara who worked on the WebNet system. Our knowledge of LAN design benefited from our domain investigators John Rieman and Ken Anderson and local informants Kyle Kucson and Evi Nemeth. The more recent research on CIEs for knowledge construction involves the author with Rogerio dePaula, Thomas Herrmann and his students at Dortmund, Ted Habermann and his group at NOAA, Dan Kowal and his middle school students, the collaborators in the 'Readings in Cognitive Science' seminar, and the researchers in the 'Articulate Learners' project. The work reported here was supported in part by grants from ARPA N66001-94-C-6038, NSF IRI-9711951, the McDonnell Foundation and NSF EAR-9870934. NetSuite Advanced Professional Design is a trademark of NetSuite.

References

- Ackerman, M.S. and McDonald, D.W. (1996). Answer Garden 2: Merging Organizational Memory with Collaborative Help, *CSCW '96*, ACM, Boston, MA. 97–105.
- Argyris, C. and Schön, D.A. (1978). *Organizational Learning: A Theory of Action Perspective*. Addison-Wesley, Reading, MA.
- Bakhtin, M. (1986). *Speech Genres and Other Late Essays*. University of Texas Press, Austin, TX.
- Blumenfeld, P., Soloway, E., Marx, R.J.K., Guzdial, M. and Palincsar, A. (1991). Motivating Project-Based Learning: Sustaining the Doing, Supporting the Learning, *Educational Psychologist*, **26** 369–398.
- Boland, R.J. and Tenkasi, R.V. (1995). Perspective Making and Perspective Taking in Communities of Knowing, *Organization Science*, **6**(4). 350–372.
- Borghoff, U. and Parechi, R. (1998). *Information Technology for Knowledge Management*. Springer, Berlin.
- Bourdieu, P. (1972). *Outline of a Theory of Practice*. Cambridge University Press, Cambridge, UK.
- Brown, J.S. and Duguid, P. (1991). Organizational Learning and Communities-of-Practice: Toward a Unified View of Working, Learning, and Innovation, *Organization Science*, **2**(1). 40–57.
- Buckingham Shum, S. (1998). Negotiating the Construction of Organizational Memories. In Borghoff, U. and Parechi, R. (eds) *Information Technology for Knowledge Management*. Springer, Berlin, 55–78.
- Buckingham Shum, S. and Hammond, N. (1994). Argumentation-Based Design Rationale: What Use at What Cost?, *International Journal of Human-Computer Studies*, **40**(4). 603–652.
- Bush, V. (1945). As We May Think, *Atlantic Monthly*, **176**(1), 101–108.
- Crook, C. (1994). *Computers and the Collaborative Experience of Learning*. Routledge, London.
- Donald, M. (1991). *Origins of the Modern Mind: Three Stages in the Evolution of Culture and Cognition*. Harvard University Press, Cambridge, MA.
- Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G. and Sumner, T. (1993). Embedding Computer-Based Critics in the Contexts of Design. In *Conference on Human Factors in Computing Systems (INTERCHI '93)*, Addison Wesley, Amsterdam. 157–164.
- Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G. and Sumner, T. (1993/1998). Embedding Critics in Design Environments. In Maybury, M. and Wahlster, W. (eds) *Readings in Intelligent User Interfaces*. Morgan-Kaufmann, San Francisco, CA, 537–561.
- Fischer, G., Lemke, A.C., McCall, R. and Morch, A. (1996). Making Argumentation Serve Design. In Moran, T. and Carrol, J. (eds) *Design Rationale: Concepts, Techniques, and Use*. Erlbaum, Mahwah, NJ, 267–293.

- Fischer, G., Grudin, J., McCall, R. et al. (1999). Seeding, Evolutionary Growth and Reseeding: The Incremental Development of Collaborative Design Environments. In Olson, G.M., Malone, T.W. and Smith, J.B. (eds) *Coordination Theory and Collaboration Technology*. Erlbaum, Mahwah, NJ.
- Floyd, C. (1992). Software Development and Reality Construction. In Floyd, C., Zuellinghoven, H., Budde, R. and Keil-Slawik, R. (eds) *Software Development and Reality Construction*. Springer, Berlin, 86–100.
- Fu, M.C., Hayes, C.C. and East, E.W. (1997). SEDAR: Expert Critiquing System for Flat and Low-Slope Roof Design and Review, *Journal of Computing in Civil Engineering*. **11**(1). 60–68.
- Gadamer, H.-G. (1960/1988). *Truth and Method*. Crossroads, New York.
- Giddens, A. (1984). *The Constitution of Society*. University of California Press, Berkeley, CA.
- Grudin, J. (1990). Why CSCW Applications Fail: Problems in Design and Evaluation, *CSCW '90*, Los Angeles, ACM. 85–93.
- Habermas, J. (1981). *Theorie des kommunikativen Handelns*, Vol. 1. Suhrkamp, Berlin.
- Heidegger, M. (1927/1996). *Being and Time: A Translation of Sein und Zeit* (Stambaugh, J., trans.). SUNY Press, Albany, NY.
- Koschmann, T. (ed.) (1996). *CSCL: Theory and Practice of an Emerging Paradigm*. Erlbaum, Hillsdale, NJ.
- Landauer, T.K. and Dumais, S.T. (1997). A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation Of Knowledge, *Psychological Review*. **104**(2). 211–240.
- Lave, J. and Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*, Cambridge University Press, Cambridge, UK.
- Lindstaedt, S. (1996). Towards Organizational Learning: Growing Group Memories in the Workplace, *CHI '96, Doctoral Consortium*, Vancouver, Addison Wesley. 14–18.
- Lindstaedt, S. and Schneider, K. (1997). Bridging the Gap Between Face-to-Face Communication and Long-Term Collaboration. In *International ACM SigGroup Conference on Supporting Group Work*, Phoenix, ACM. 331–340.
- Marx, K. (1867/1976). *Capital*, Vol. I (Fowkes, B., trans.). Penguin, New York.
- McCall, R., Bennett, P., d'Oronzio, P., Ostwald, J., Shipman, F. and Wallace, N. (1990). PHIDIAS: Integrating CAD Graphics into Dynamic Hypertext. In A. Rizk, N. Streitz, J. Andre (eds) *Hypertext: Concepts, Systems and Applications*. Cambridge U. Press, Cambridge. 152–165.
- Miller, P.L. (1986). *Expert Critiquing Systems: Practice-Based Medical Consultation by Computer*. Springer, Berlin.
- Mittal, S., Bobrow, D. and Kahn, K. (1986). Virtual Copies at the Boundary Between Classes and Instances. In *Object-Oriented Programming Systems, Languages and Applications (OOPSLA '86)*, 159–166. Portland, ACM.
- Nelson, T. (1981). *Literary Machines*. Mindful Press, New York.
- Norman, D.A. (1993). *Things That Make Us Smart*. Addison-Wesley, Reading, MA.
- O'Malley, C. (1995). *Computer Supported Collaborative Learning*. Springer, Berlin.
- Orlikowski, W. (1992). Learning from Notes: Organizational issues in groupware Implementation. In *CSCW '92*, Toronto, ACM. 362–369.
- Orlikowski, W., Yates, Y., Okamura, K. and Fujimoto, M. (1995). Shaping Electronic Communication: The Metastructuring of Technology in the Context of Use, *Organization Science*. **6**(4). 423–369.
- Orr, J. (1990). Sharing Knowledge, Celebrating Identity: War Stories and Community Memory in a Service Culture. In Middleton, D.S. and Edwards, D. (eds) *Collective Remembering: Memory in Society*. Sage, Beverly Hills, CA.
- Polanyi, M. (1962). *Personal Knowledge*. Routledge & Kegan Paul, London.
- Repenning, A. (1994). Programming Substrates to Create Interactive Learning Environments, *Journal of Interactive Learning Environments, Special Issue on End-User Environments*. **4**(1). 45–74.
- Rittel, H. and Webber, M.M. (1984). Planning Problems Are Wicked Problems. In Cross, N. (ed.) *Developments in Design Methodology*. Wiley, New York, 135–144.
- Robbins, J.E. and Redmiles, D.F. (1998). Software Architecture Critics in the Argo Design Environment, *Knowledge-Based Systems* pp. 47–60.
- Scardamalia, M. and Bereiter, C. (1991). Higher Levels of Agency in Knowledge Building: A Challenge for the Design of New Knowledge Media, *Journal of the Learning Sciences*. **1**. 37–68.
- Scardamalia, M. and Bereiter, C. (1996). Computer Support for Knowledge-Building Communities. In Koschmann, T. (ed.) *CSCL: Computer Supported Collaborative Learning*. Erlbaum, Hillsdale, NJ, 249–268.
- Schön, D.A. (1983). *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, New York.
- Senge, P. (1990). *The Fifth Discipline: The Art and Practice of the Learning Organization*. Currency Doubleday, New York.
- Stahl, G. (1993a). *Interpretation in Design: The Problem of Tacit and Explicit Understanding in Computer Support of Cooperative Design*. Ph.D. Dissertation, Technical Report No. CU-CS-688-93, Department of Computer Science, University of Colorado, Boulder, CO.

- Stahl, G. (1993b). Supporting Situated Interpretation. In *Proceedings of the Cognitive Science Society (CogSci '93)*, Boulder, CO, Lawrence Erlbaum. 965–970.
- Stahl, G. (1995). *Supporting Personalizable Learning*. Technical Report No. CU-CS-788-95, Department of Computer Science, University of Colorado, Boulder, CO.
- Stahl, G. (1998). WebGuide: Guiding Collaborative Learning on the Web with Perspectives. In *AERA '99*. Montreal, ERIC.
- Stahl, G., Sumner, T. and Owen, R. (1995). Share Globally, Adapt Locally: Software to Create and Distribute Student-Centered Curriculum, *Computers and Education. Special Issue on Education and the Internet*. **24**(3). 237–246.
- Tomasello, M., Kruger, A.C. and Ratner, H. (1993). Cultural Learning, *Behavioral and Brain Sciences*. 16. 495–552.
- Vygotsky, L. (1930/1978). *Mind in Society*. Harvard University Press, Cambridge, MA.
- Winograd, T. and Flores, F. (1986). *Understanding Computers and Cognition: A New Foundation of Design*. Addison-Wesley, Reading, MA.
- Yates, J. and Orlikowski, W. (1992). Genres of Organizational Communication: A Structural Approach to Studying Communication and Media, *Academy of Management Review*. **17**(2). 299–326.
- Zuboff, S. (1988). *In the Age of the Smart Machine*. Basic Books, New York.

Correspondence and offprint requests to: Gerry Stahl, Centre for LifeLong Learning and Design and Institute of Cognitive Science, University of Colorado at Boulder, CO 80309, USA. Email: gerry@cs.colorado.edu