Selected Writings

Gerry Stahl

March 16, 2000

1. Stahl, G. (1993) Supporting situated interpretation, In: *Proceedings of Annual Meeting of the Cognitive Science Society* (CogSci '93), Boulder, CO, pp. 965-970.

2. Stahl, G. (1996) Armchair missions to Mars: Using case-based reasoning and fuzzy logic to simulate a time series model of astronaut crews, *Knowledge-Based Systems*, 9, pp. 409-415. Forthcoming in Sankar K. Pal, Daniel Yeung So, Tharam Dillon (Eds.) (2000) *Soft Computing in Case Based Reasoning*, Springer Verlag.

3. Stahl, G., Sumner, T., & 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), pp. 237-246.

4. Stahl, G. (2000) Collaborative information environments to support knowledge construction by communities, Forthcoming in: *AI & Society*.

5. Stahl, G. (1999) WebGuide: Guiding collaborative learning on the Web with perspectives, Presented at: *Annual Conference of the American Educational Research Association* (AERA '99), Montreal, Canada.

6. Stahl, G. (2000) Perspectives on collaborative knowledge-building. Proposal to National Science Foundation program in Information Technology Research.

INTRODUCTION

The writings assembled here display a cross-section of my work in computer science and cognitive science during the 1990's. They also provide a glimpse into my plans for the 2000's.

1. "*Supporting situated interpretation*" encapsulates my dissertation research. It presents a philosophy of situated interpretation that draws upon the hermeneutic philosophy of my earlier studies in the 1970's (Heidegger, Gadamer, Habermas) and more recent work by others in situated cognition (Suchman, Winograd & Flores). These ideas are developed in the context of design theory (Rittel, Alexander, Schön). Consequences are drawn for computer support of design, and embedded in HERMES, a software prototype. HERMES features: (a) a hypermedia structure for drawings, design rationale, and critic agents; (b) an end-user extensible programming language for defining displays and critics; and (c) a computational perspectives mechanism.

2. "*Armchair missions to Mars: Using case-based reasoning and fuzzy logic to simulate a time series model of astronaut crews*" reports on consulting work with NASA after graduation. The CREW system predicts outcomes for a given long-term space mission based on psychological characteristics of a selected crew. The development of my CREW system involved the definition, extension, and integration of several artificial intelligence algorithms.

3. "*Share globally, adapt locally: Software to create and deliver student-centered curriculum*" defines the Teachers' Curriculum Assistant (TCA) that I designed and prototyped. This project reflected my decision to focus on educational technology and to try to realize some of the elusive potential of the then-new Web for personalizable learning. The goal was to provide: (a) a digital library of materials for constructivist learning activities; (b) a forum for teachers to share their versions of these materials, their curricula incorporating them, and their experiences using them; and (c) mechanisms for personalizing the materials to specific classrooms and individual learners.

4. "*Collaborative information environments to support knowledge construction by communities*" traces an historical shift in my software design approaches. Systems like HERMES were domain-oriented design environments (DODEs) that took advantage of embedded domain knowledge to support the work of individual designers. Spurred by the affordances and vision of the Web, collaborative information environments (CIEs) like TCA and WEBNET stressed the ability of distributed communities of practice to evolve their own bodies of organizational learning and to make their organizational memories persistent and accessible. This approach then led to collaborative knowledge-building environments (KBEs) like WEBGUIDE to support classroom as well as workplace learning.

5. "*WebGuide: Guiding collaborative learning on the Web with perspectives*" reports on experiences using WEBGUIDE – my KBE prototype – in a middle school environmental science course and in a cognitive science graduate seminar. These experiences contributed to my theory of KBEs as well as to an understanding of the social informatics practicalities involved in deploying such systems. In particular, WEBGUIDE explores the use of networks of computational perspectives to manage personal and group interpretations of shared knowledge in a Web repository.

6. "*Perspectives on collaborative knowledge-building*" proposes a research agenda to pursue work on KBEs like WEBGUIDE. This includes: (a) theory of computer-supported collaborative learning; (b) software component architectures for rapid prototyping of KBEs; (c) deployment and observation of KBEs in a variety of usage settings; and (d) collaboration with other research groups working on KBEs in the US and abroad. The long-range goal is to achieve a new stage of human cognition through technology-mediated collaboration.

These six articles are representative of my writings. A resume including a list of publications is appended. Most of my writings are available at: www.cs.colorado.edu/~gerry/publications.

Gerry Stahl
Boulder, CO
March 16, 2000

CONTENTS

SUPPORTING SITUATED INTERPRETATION

ABSTRACT

This paper discusses the role of interpretation in innovative design and proposes an approach to providing computer support for interpretation in design.

According to situated cognition theory, most of a designer's knowledge is normally tacit. Situated interpretation is the process of explicating something that is tacitly understood, within its larger context.

The centrality of interpretation to non-routine design is demonstrated by: a review of the design methodology of Alexander, Rittel, and Schön; a protocol analysis of a lunar habitat design session; and a summary of Heidegger's philosophy of interpretation. These show that the designer's articulation of tacit knowledge takes place on the basis of an understanding of the design situation, a focus from a particular perspective, and a shared language.

As knowledge is made explicit through the interpretive processes of design it can be captured for use in computer-based design support systems. A prototype software system is described for representing design situations, interpretive perspectives, and domain terminology to support interpretation by designers.

THE NEED FOR COMPUTER SUPPORT

The volume of information available to people is increasing rapidly. For many professionals this means that the execution of their jobs requires taking into account far more information than they can possibly keep in mind. Consider the lunar habitat designers who serve as a key example in this paper. In working on their high-tech design tasks, they must take into account architectural knowledge, ergonomics, space science, NASA regulations, and lessons learned in past missions. Computers seem necessary to store these large amounts of data. However, the problem is how to capture and encode information relevant to novel future tasks and how to present it to designers in formats that support their mode of work.

A framework for clarifying the respective roles for computers and people in tasks like lunar habitat design is suggested by the theory of *situated cognition*. Several influential recent books (e.g., Schön, 1983; Winograd & Flores, 1986; Suchman, 1987; Ehn, 1988; Dreyfus, 1991) argue that human cognition is fundamentally different from computer manipulations of formal symbol systems. These differences imply that people need to retain control of the processes of non-routine design, but that computers can provide valuable computational, visualization, and external memory aids for the designers and support interpretation by them.

From the viewpoint of situated cognition, the greatest impediment to computer support of innovative design is that designers make extensive use of *tacit* knowledge while computers can only use *explicit* representations of information. This paper discusses the role of tacit understanding in designing, in order to motivate an approach to computer support of design. It focuses on three themes: (a) the need to represent novel design *situations*; (b) the importance of viewing designs from multiple *perspectives*; and (c) the utility of formulating tacit knowledge in explicit *language*.

The following sections discuss how these three themes figure prominently in analyses of interpretation in design methodology and in a study of interpretation in lunar habitat design. Following a discussion of the tacit basis of understanding, the philosophy of interpretation defines interpretation as the articulation of tacit understanding. Then consequences for computer support for interpretation are drawn, and they are illustrated by the HERMES system, a prototype for supporting interpretation in the illustrative task of lunar habitat design.

INTERPRETATION IN DESIGN METHODOLOGY

The centrality of interpretation to design can be seen in seminal writings of design methodologists. The following summaries highlight the roles of appropriate representations of the design situation, alternative perspectives, and linguistic explications of tacit understanding within the processes of interpretation in design.

Alexander (1964) pioneered the use of computers for designing. He used them to compute diagrams or patterns that decomposed the structural dependencies of a given problem into relatively independent substructures. In this way, he developed explicit interpretations for understanding a task based on an analysis of the unique design *situation*.

For Rittel (1973), the heart of design is the deliberation of issues from multiple *perspectives*. Interpretation in design is "an argumentative process in the course of which an image of the problem and of the solution emerges gradually among the participants, as a product of incessant judgment, subjected to critical argument" (p.162). Rittel's idea of using computers to keep track of the various issues at stake and alternative positions on those issues led to the creation of issue-based information systems.

Schön (1983) argues that designers constantly shift perspectives on a problem by bringing various professionally trained tacit skills to bear, such as visual perception, graphical sketching, and vicarious simulation. By experimenting with tentative design moves within the tacitly understood situation, the designer discovers consequences and makes aspects of the structure of the problem explicit. Certain features of the situation come into focus and can be named or characterized in *language*. As focus subsequently shifts, what has been interpreted may slip back into an understanding that is once more tacit, but is now more developed.

Interpretation in Lunar Habitat Design

As part of an effort at developing computer support for lunar habitat designers, thirty hours of design sessions were videotaped and analyzed (see Stahl, forthcoming). The specified task was to accommodate four astronauts for 45 days on the moon in a cylindrical module 23 feet long and 14 feet wide.

A protocol analysis of the designers' activities shows that much of the design time consisted of processes of *interpretation*, i.e., the explication of previously tacit understanding. As part of this interpretation, representations were developed for describing pivotal features of the design situation that had not been included in the original specification; perspectives were evolved for looking at the task; and terminology was defined for explicitly naming, describing, and communicating shared understandings.

The designers felt that a careful balance of public and private space would be essential given the crew's long-term isolation in the habitat. An early design sketch proposed private crew areas consisting of a bunk above a workspace for each astronaut. Space constraints argued against this. The traditional conception of private space as a place for one person to get away was made explicit and criticized as taking up too much room. As part of the interpretive designing process, this concept was revised into a reinterpretation of privacy as a gradient along the habitat from quiet sleep quarters to a public activity area. This notion of degrees of privacy permitted greater flexibility in designing.

In another interchange related to privacy, the conventional American idea of a bathroom was subjected to critical deliberation when it was realized that the placement of the toilet and that of the shower were subject to different sets of constraints based on life in the habitat. The tacit acceptance of the location of the toilet and shower together was made explicit by comparing it to alternative European perspectives. The revised conception permitting a separation of the toilet from the shower facilitated a major design reorganization.

In these and other examples, the designers needed to revise their representations for understanding the design *situation*. They went from looking at privacy as a matter of individual space to reinterpreting the whole interior space as a continuum of private to public areas.

The conventional American notion of a bathroom was compared with other cultural models and broken down into separable functions that could relate differently to habitat usage patterns. Various *perspectives* were applied to the problem, suggesting new possibilities and considerations. Through discussion, the individual perspectives merged and novel solutions emerged.

In this interpretive process, previously tacit features of the design became explicit by being named and described in the *language* that developed. For instance, the fact that quiet activities were being grouped toward one end of the habitat design and interactive ones at the other became a topic of conversation at one point and the terminology of a "privacy gradient" was proposed to clarify this emergent pattern.

The Tacit Basis of Understanding

Situated cognition theory disputes the prevalent view that all human cognition is based on explicit mental representations such as goals and plans. Winograd and Flores (1986) hold that "experts do not need to have formalized representations in order to act" (p.99). Although manipulation of such representations is often useful, there is a background of preunderstanding that cannot be fully formalized as explicit symbolic representations subject to rule-governed manipulation. This tacit preunderstanding underlies people's ability to understand representations when they do make use of them. Suchman (1987) concurs that goals and plans are secondary phenomena in human behavior, usually arising only after action has been initiated: "when situated action becomes in some way problematic, rules and procedures are explicated for purposes of deliberation and the action, which is otherwise neither rule-based nor procedural, is then made accountable to them" (p.54).

Philosophers like Polanyi (1962), Searle (1980), and Dreyfus (1991) suggest a variety of reasons why tacit preunderstanding cannot be fully formalized as data for computation. First, it is too vast: background knowledge includes bodily skills and social practices that result from immense histories of life experience and that are generally transparent to us. Second, it must be tacit to function: we cannot formulate, understand, or use explicit knowledge except on the basis of necessarily tacit preunderstandings.

This is not to denigrate conceptual reasoning and rational planning. Rather, it is to point out that the manipulation of formal representations alone cannot provide a complete model of human understanding. Rational thought is an advanced form of cognition that distinguishes humans. Accordingly, an evolutionary theorist of consciousness such as Donald (1991) traces the development of symbolic thought from earlier developmental stages of tacit knowing, showing how these earlier levels persist in rational human thought as the necessary foundation for advanced developments, including language, writing, and computer usage.

The most thorough formulation of a philosophical foundation for situated cognition theory is given by Heidegger (1927), the first to point out the role of tacit preunderstanding and to elaborate its implications. For Heidegger, we are always knowledgeably embedded in our world; things of concern in our situations are already meaningful in general before we engage in cognitive activity. We know how to behave without having to think about it. For instance, an architect designing a lunar habitat knows how to lift a pencil and sketch a line, or how to look at a drawing and see the rough relationships of various spaces pictured there. The architect understands what it is to be a designer, to critique a drawing, to imagine being a person walking through the spaces of a floor plan.

Heidegger defines the *situation* as the architect's context—including the physical surroundings, the available tools, the circumstances surrounding the task at hand, and the architect's own personal or professional aims. The situation constitutes a network of significance in terms of which each part of the situation is already meaningful (see Stahl, 1975). That is, the architect has tacit knowledge of the situation as a whole; if something becomes a focus for the architect, it is perceived as already understood and its meaning is defined by its relations to the rest of the situation.

To the architect, a rectangular arrangement of lines on a piece of paper is not perceived as meaningless lines, but, given the design situation, it is already understood as a bunk for astronauts. The bunk is implicitly defined as such by the design task, the shared intentions of the design team, the other elements of the design, the sense of space conveyed by the design, and so on indefinitely. This network of significance is background knowledge that allows the architect to think about features of the design, to make plans for changes, and to discover problems or opportunities in the evolving design. At any given moment, the background is already tacitly understood and does not need to be an object of rational thought manipulating symbolic representations.

At some point the architect might realize that the bunk is too close to a source of potential noise, like the flushing of the toilet. The explicit concern about this physical adjacency arises and becomes something important against the background of relationships of the preunderstood situation. Whereas a commonsensical view might claim that the bunk and toilet were already present and therefore their adjacency was always there by logical implication, Heidegger proposes a more complex reality in which things are ordinarily hidden from explicit concern. In various ways, they can become uncovered and discovered, only to re-submerge soon into the background as our focus moves on.

In this way, our knowledge of the world does not consist primarily in mental models that represent an objective reality. Rather, our understanding of things presupposes a tacit preunderstanding of our situation. Only as situated in our already interpreted world can we discover things and construct meaningful representations of them. Situated cognition is not a simplistic theory that claims our knowledge lies in our physical environment like words on a sign post: it is a sophisticated philosophy of interpretation.

The Philosophy of Interpretation

Human understanding develops through interpretive explication. According to Heidegger, interpretation provides the path from tacit, uncritical preunderstandings to reflection, refinement, and creativity. The structure of this process of interpretation reflects the inextricable coupling of the interpreter with the situation, i.e., of people with their worlds. Our situation is not reducible to our preunderstanding of it; it offers untold surprises, which may call for reflection, but which can only be discovered and comprehended thanks to our preunderstanding. Often, these surprise occasions signal *breakdowns* in our skillful, transparent behavior, although we can also make unexpected discoveries in the situation through conversation, exploration, natural events, and other occurrences.

A discovery breaks out of the preunderstood situation because it violates or goes beyond the network of tacit meanings that make up the preunderstanding of the situation. To understand what we have discovered, we must explicitly *interpret* it *as* something, as having a certain significance, as somehow fitting into the already understood background. Then it can merge into our comprehension of the meaningful situation and become part of the new background. Interpretation of something *as* something is always a reinterpretation of the situated context.

For instance, the lunar habitat designers discovered problems in their early sketches that they interpreted as issues of privacy. Although they had created the sketches themselves, they were completely surprised to discover certain conflicts among the interactions of adjacent components, like the bunks and the toilet. Of course, the discoveries could only occur because of their understanding of the *situation*, represented in their drawings. The designers paused in their sketching to discuss the new issues. First they debated the matter from various *perspectives*: experiences of previous space missions, cultural variations in bathroom designs, technical acoustical considerations. Then they considered alternative conceptions of privacy, gradually developing a shared *vocabulary* that guided their revisions and became part of their interpretation of their task. They reinterpreted their understanding of privacy and represented their new view as a "privacy gradient."

These themes of representing the situation, changing perspectives, and using explicit language correspond to the three-fold structure of interpretation in Heidegger's philosophy. He articulates the preconditions of interpretation as: (a) *prepossession* of the situation as a network of preunderstood significance; (b) *preview* or expectations of things in the world as being structured in certain ways; and (c) *preconception*, a language for expressing and communicating.

In other words, interpretation never starts from scratch or from an arbitrary assignment of representations, but is an evolving of tentative preunderstandings and anticipations. One necessarily starts with sets of "prejudices" that have been handed down historically; the interpretive process allows one to reflect upon these preunderstandings methodically and to refine new meanings, perspectives, and terminologies for understanding things more appropriately.

Computer Support for Interpretation

The theory of situated cognition and the philosophy of interpretation stress how different human understanding is from computer manipulations of arbitrary symbols. These theories suggest the approach of *augmenting* (rather than automating) human intelligence. According to this approach, software can at best provide computer representations for people to interpret based on their tacit understanding of what is represented.

Representations used in computer programs must be carefully structured by human programmers who understand the task being handled thoroughly, because the computer itself simply follows the rules it has been given for manipulating symbols, with no notion of what these symbols represent. People who understand the domain must codify their background knowledge into software rules sufficiently to make the computer algorithms generate results that will be judged correct when interpreted by people. Only if a domain can be strictly delimited and its associated knowledge exhaustively reduced to rules, can it be completely automated.

Many tasks like lunar habitat design that call for computer support do not have such strictly delimited domains with fully catalogued and formalized knowledge bases. These domains may require exploration of problems never before considered, assumption of creative viewpoints, or formulation of innovative concepts. Software to support designers in such tasks should provide facilities for the creation of new representations and flexible modification of old ones. As the discussion of Alexander emphasized, the ability to develop appropriate representations dynamically is critical. Because they capture understandings of the *situation* that evolve through processes of interpretation, representations need to be modifiable during the design process itself and cannot adequately be anticipated in advance or provided once and for all.

The concept of an objective, coherent body of domain knowledge is misleading. As Rittel said, non-routine design is an argumentative process involving the interplay of unlimited perspectives, reflecting differing and potentially conflicting technical concerns, personal idiosyncrasies, and political interests. Software to support design should capture these alternative deliberations on important issues, as well as document specific solutions. Furthermore, because all design knowledge may be relative to perspectives, the computer should be used to define a network of over-lapping *perspectives* by which to organize issues, rationale, sketches, component parts, and terminology.

As Schön emphasized, interpretive design relies on moving from tacit skills to explicit conceptualizations. Additionally, design work is inherently communicative and increasingly collaborative, with high-tech designs requiring successive teams of designers, implementers, and maintainers. Software to support collaborative design should provide a *language* facility for designers to develop a formal vocabulary for expressing their ideas, for communicating them to future collaborators, and for formally representing them within computer-executable software. An end-user language is needed that provides an extensible domain vocabulary, is usable by non-programmers, and encourages reuse and modification of expressions.

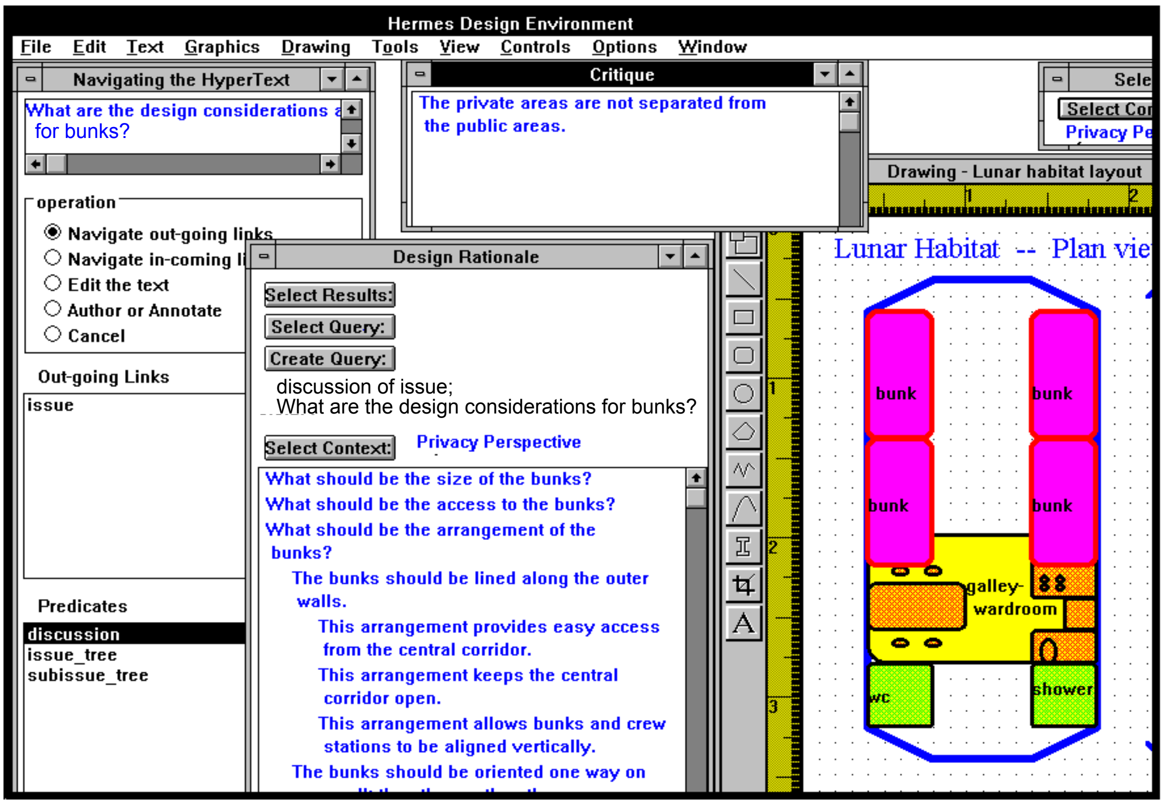


Figure 1. A view of the HERMES design environment, showing (left to right) a dialogue for browsing, a view of the issue base, a critic message, a construction area, and a button for changing interpretive perspectives.

Heidegger's analysis of interpretation suggests that most of the information that would be useful to designers may be made explicit at some moment of interpretation during designing. One strategy for accumulating a useful knowledge base is to have the software capture knowledge that becomes explicit while the software is being used. As successive lunar habitats are designed on a system, issues and alternative deliberations can accumulate in its issue base; new perspectives can be defined containing their own modifications of terminology and critic rules; the language can be expanded to include more domain vocabulary, conditional expressions, and query formulations. In this way, potentially relevant information is captured in formats useful for designers, because it is a product of human interpretation.

This is an evolutionary, bootstrap approach, where the software can not only support individual design projects, but simultaneously facilitate the accumulation of expertise and viewpoints in open-ended, exploratory domains. This means that the software should make it easy for designers to formalize their knowledge as it becomes explicit, without requiring excessive additional effort. The software should reward its users for increasing the computer knowledge base by performing useful tasks with the new information, like providing documentation, communicating rationale, and facilitating reuse and modification of relevant knowledge.

THE HERMES SYSTEM

In Greek mythology, Hermes supported human interpretation by providing the gift of spoken and written language and by delivering the messages of the gods. A prototype software system named HERMES has been designed to support the preconditions of interpretation (a) by representing the design construction situation for *prepossession*, (b) by providing alternative perspectives for *preview*, and (c) by including an end-user language for *preconception*.

It supports tacit knowing by encapsulating (a) mechanisms for analyzing design situations using interpretive critics (Fischer, et al., 1993), (b) alternative sets of information organized in named perspectives (Stahl, 1993), and (c) hypermedia computations expressed in language terms (Stahl, et al., 1992). In each of these cases, the hidden complexities can be made explicit upon demand, so the designer can reflect upon the information and modify (reinterpret) it.

HERMES is a knowledge-representation substrate for building computer-based design assistants (like that in Figure 1). It provides various media for designers to build formal representations of design knowledge. The hypermedia network of knowledge corresponds to the design *situation*. Nodes of the knowledge representation can be textual statements for the issue base, CAD graphics for sketches, bitmap images to illustrate ideas, sound for audio commentary, or language expressions for critics and queries.

HERMES supports the collaborative nature of design by multiple teams through its *perspectives* mechanism. This allows users to organize knowledge in the system into over-lapping collections. Drawings, definitions of domain terms in the language, computations for critic rules, and annotations in the issue base can be grouped together for a project, a technical specialty, an individual, a team, or an historical version. Every action in HERMES takes place within some defined perspective, which determines what versions of information are currently accessible.

The HERMES *language* pervades the system, defining mechanisms for browsing, displaying, and critiquing all information. This means that designers can refine the representations, views, and expressions of all forms of domain knowledge in the system. Vocabulary in the language is modifiable and every expression can be encapsulated by a name. The syntax is English-like, in an effort to make statements in the language easily interpretable. The language is declarative, so users need not be bothered with explicit sequential programming concerns. Combined with the perspectives mechanism, the language permits designers to define and refine their own interpretations. This allows the HERMES substrate to support multiple situated interpretations.

CONCLUSION

The theory of situated cognition argues that only people's tacit preunderstanding can make data meaningful in context. Neither people nor computers alone can take advantage of huge stores of data; such information is valueless unless designers use it in their interpretations of design situations. The data handling capabilities of computers should be used to support the uniquely human ability to understand. The philosophy of interpretation suggests that several aspects of human understanding and collaboration can be supported with mechanisms like those in HERMES for refining representations of the design situation, for creating alternative perspectives on the task, and for sharing linguistic expressions. Together, situated cognition theory and Heidegger's philosophy of interpretation provide a theoretical framework for a principled approach to computer support for designers' situated interpretation in the information age.

ACKNOWLEDGMENTS

This paper is indebted to Gerhard Fischer, Ray McCall, Kumiyo Nakakoji, Jonathan Ostwald, and Tamara Sumner, who helped to focus it, and to the HCC research group at the University of Colorado generally, whose ideas and systems it tries to ground theoretically and extend practically.

REFERENCES

Alexander C (1964) *Notes on the Synthesis of Form*. Cambridge: Harvard University Press.

Donald M (1991) Origins of the Modern Mind: Three Stages in the Evolution of Culture and Cognition. Cambridge: Harvard University Press.

Dreyfus H (1991) Being-in-the-World: A Commentary on Heidegger's Being and Time, Division I. Cambridge: MIT Press.

Ehn P (1988) *Work-Oriented Design of Computer Artifacts*. Stockholm: Arbetslivscentrum.

Fischer G, Nakakoji K, Ostwald J, Stahl G, Sumner T (1993) Embedding Computer-Based Critics in the Contexts of Design. In *Proc. of InterCHI* '93.

Heidegger M (1927) *Sein und Zeit*. Tübingen: Niemeyer. Translation: Heidegger M (1962) *Being and Time*. New York: Harper & Row.

Polanyi M (1962) *Personal Knowledge.* London: Routledge & Kegan Paul.

Rittel H, Webber M (1973) Dilemmas in a General Theory of Planning. *Policy Science*. *4*: 155-169.

Searle J (1980) Minds, Brains, and Programs. *The Behavioral and Brain Sciences*. *3*.

Schön D (1983) *The Reflective Practitioner*. New York: Basic Books.

Stahl G (1975) *Hermeneutics and Social Theory: Interpreting and Transforming Our World*. Ph.D. Diss., Dept. of Philosophy, Northwestern Univ.

Stahl G (1993) Supporting Interpretation in Design. *Journal of Architecture and Planning Research*.

Stahl G (forthcoming*) Interpretation in Design: The Problem of Tacit and Explicit Understanding in Computer Support of Cooperative Design*. Ph.D. Diss., Dept. of Computer Science, Univ. of Colorado.

Stahl G, McCall R, Peper G (1992) Extending Hypermedia with an Inference Language: an Alternative to Rule-Based Expert Systems. In *Proc. of the IBM ITL Conference: Expert Systems (October 19-21, 1992).*

Suchman L (1987) *Plans and Situated Actions: the Problem of Human Machine Communication*. Cambridge: Cambridge University Press.

Winograd T, Flores F (1986) *Understanding Computers and Cognition: A New Foundation for Design*. New York: Addison-Wesley.

ARMCHAIR MISSIONS TO MARS: USING CASE-BASED REASONING AND FUZZY LOGIC TO SIMULATE A TIME SERIES MODEL OF ASTRONAUT CREWS

ABSTRACT

Computer simulation of long missions in space can provide experience and predictions without the expense and risk of actual flights. Simulations are most helpful if they can model the behavior of key psychological factors of the crew over time, rather than simply predicting overall mission success. Because of the lack of experience with interplanetary trips and the problems of generalizing and adapting data on analog missions, it is not possible to formulate a set of formal rules adequate for building an expert system. Rather, a case-based reasoning approach to constructing a time series model is pursued. Even for this approach, however, the case base must be supplemented by adaptation rules. These rules of thumb are gleaned from the social science literature on small group interactions under extreme conditions of isolation and confinement. The non-quantitative nature of these rules lends itself to formulation and computation using fuzzy logic. The application domain presents several technical issues for traditional case-based reasoning: there is no natural hierarchy of parameters to use in optimizing installation and retrieval of cases, and there are large variations in behavior among similar missions. These problems are addressed by custom algorithms to keep the computations tractable and plausible. Thus, the harnessing of case-based reasoning for this practical application requires the crafting of a custom, hybrid system.

Keywords: simulation, case-based reasoning, statistical modeling

PREFACE

Background of the Research

During the period of space exploration around 1993, planners at NASA (the US space agency) were concerned about interpersonal issues in astronaut crew composition. The nature of astronaut crews was undergoing significant change. In the past, astronauts were primarily young American males with rigorous military training; missions were short; crews were small. Prior to a mission, a crew trained together for about a year, so that any interpersonal conflicts could be worked out in advance. The future, however, promised crews that would be far less homogeneous and regimented: international crews speaking different languages, mixed gender, inter-generational, larger crews, longer missions. This was the start of Soviet-American cooperation and planning for Space Station. While there was talk of a manned expedition to Mars, the more likely scenario of an international Space Station with six-month crew rotations was a realistic concern.

There was not much experience with the psychological effects on crews confined in isolated and extreme conditions for months at a time. The data from submarines and Antarctic winter-overs provided some indications, but it was limited, inappropriately documented and inconsistent. NASA was beginning to conduct some experiments where they could collect the kinds of data they needed. But they required a way of analyzing such data, generalizing it and applying it to projected scenarios.

The Soft Computing Algorithms

NASA wanted a way of predicting how a given crew – with a certain mix of astronauts – might respond to mission stress under different scenarios. This would require a complex model with many parameters. There would never be enough relevant data to derive the parameter values statistically. Given a modest set of past cases, the method of case-based reasoning suggested itself. A case-based system requires (1) a mechanism for retrieving past cases similar to a proposed new case and (2) a mechanism for adapting the data of a retrieved case to the new case based on the differences between the two.

For the retrieval mechanism, we defined a number of characteristics of astronauts and missions. The nature of our data and these characteristics raised a number of issues for retrieval and we had to develop innovative modifications of the standard case-based reasoning algorithms, as described in detail below.

For the adaptation mechanism, we developed a model of the mission based on a statistical approach known as interrupted time series analysis. In addition, we derived a set of adaptation rules based on the social science literature about confined and isolated crews. We formulated these rules in English and represented them in the software using fuzzy logic.

The Software System

We developed a case-based reasoning software system named CREW. To make the retrieval of cases tractable, scalable and efficient, we developed the system in a database programming environment. We selected FoxPro because it was highly optimized, included a general purpose programming language and was compatible with both Windows and the Macintosh.

Most of the software code consisted of the algorithms described in this chapter. Because CREW was intended to be a proof-of-concept system, its data entry routines and user interface were minimal. The user interface consisted of a set of pull-down menus for selecting a variety of testing options and a display of the results in a graph format (see Figure 1). Some of the steps in the reasoning were printed out so that one could study the reasoning process.

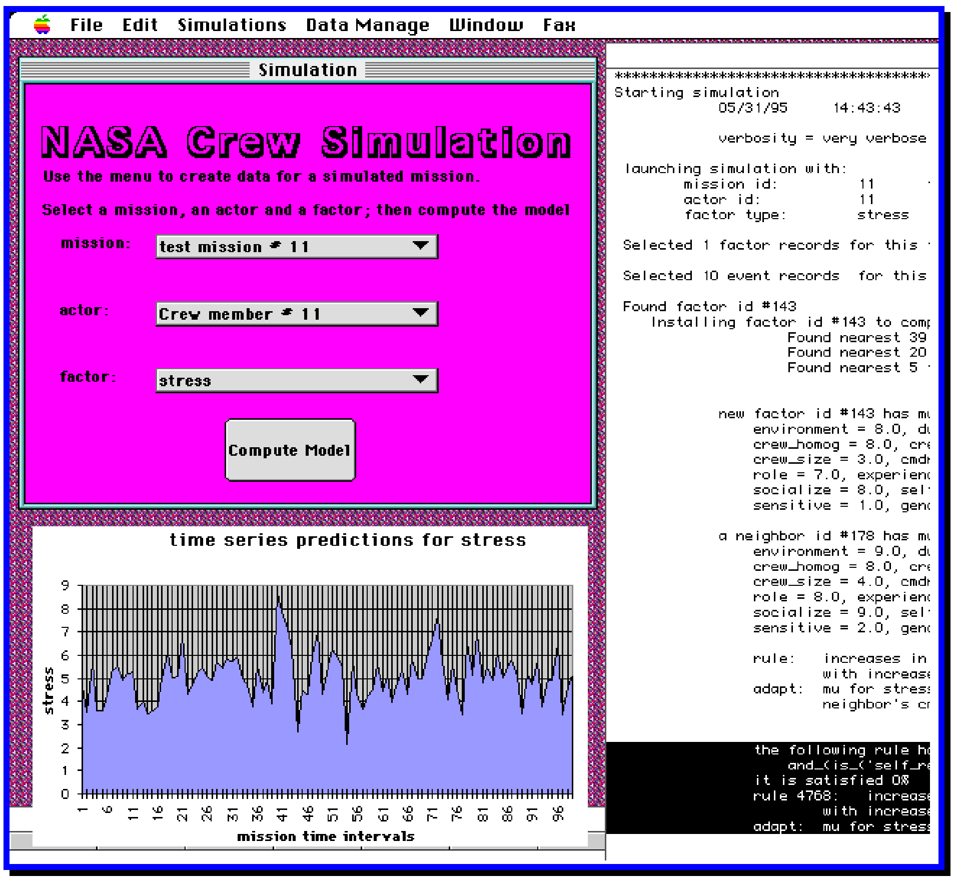


Figure 1. A view of the CREW interface. Upper left allows selection of mission characteristics. Menu allows input of data. Lower left shows magnitude of a psychological factor during 100 points in the simulated mission. To the right is a listing of some of the rules taken into account.

By the end of the project, we successfully demonstrated that the time series model, the case-based reasoning and the fuzzy logic could all work together to perform as designed. The system could be set up for specific crews and projected missions and it would produce sensible predictions quickly. The next step was to enter real data that NASA was just beginning to collect. Because of confidentiality concerns, this had to be done within NASA, so we turned over the software to them for further use and development.

The People Involved

The research was sponsored by the Behavioral and Performance Laboratory at Johnson Space Center in Houston, Texas, part of NASA's Astronaut Support Division. We worked closely with NASA researchers Dr. Joanna Wood and Dr. Albert Holland on the design of the software and the data. At the end of the project, we delivered the software to them to continue the work.

The research was conducted at Owen Research, Inc. (ORI) in Boulder, Colorado. ORI is a small research lab founded and run by Dr. Robert Owen. Owen is a physicist specializing in laser optics. He also has a Ph.D. in anthropology, and his dissertation in that field led to this research in modeling small group behavior using AI (artificial intelligence) techniques. I developed the technical approach and programmed the system. Dr. Brent Reeves assisted with the fuzzy logic algorithms. To help collect and analyze social science literature related to small groups in isolated conditions, we worked with Professor Russell McGoodwin of the Anthropology Department at the University of Colorado (CU) and his student, Nick Colmenares. In addition, I conducted several interviews of an experienced astronaut, Mike Lounge, and discussed our project with him.

I began this project immediately after completing my Ph.D. dissertation in computer science at CU, where I had specialized in AI. Since my undergraduate years at MIT in the mid-sixties and a Ph.D. in philosophy in the mid-70's, I have worked as a systems programmer, software developer and computer consultant. Following this project, I continued to work with ORI on software for an optical bench to stabilize interferometry equipment during space flight and on an Internet-based system for teachers to share curriculum ideas. After working at ORI, I returned to CU, where I am now a Research Professor in cognitive science and computer science. My current research involves Web-based environments for collaborative learning and knowledge-building – for further information or to contact me, see http://www.cs.colorado.edu/~gerry.

"Armchair missions to Mars" describes research conducted during a two year SBIR (Small Business Innovative Research) grant (Project 91-1-II-1201-9027) from NASA in 1993-1995. It was originally published (except for this Preface) in the journal *Knowledge-Based Systems* in 1996, volume 9, pages 409-415, and is reprinted below with the permission of the publisher, Elsevier.

INTRODUCTION

The prospect of a manned mission to Mars has been debated for 25 years since the first manned landing on the moon [1]. It is routinely argued that this obvious next step in human exploration is too costly and risky to undertake, particularly given our lack of experience with lengthy missions in space [2].

Social science research to explore issues of the effects of such a mission on crew members has focused on experience in *analog* missions under extreme conditions of isolation and confinement, such as Antarctic winter-overs, submarine missions, orbital space missions and deep sea experiments [3]. This research has produced few generalizable guidelines for planning a mission to Mars [4].

We have undertaken to simulate the effects of interplanetary missions in a computer program named CREW. This program is for use by NASA to assist in astronaut crew selection and mission planning [5]. Given descriptions of tentatively selected crew members and of scheduled activities, CREW simulates the mission and reports on the probable course of particular factors during the duration of the mission.

We are working with staff at the psychology labs of NASA's astronaut support division, so we have focused on psychological factors of the crew members, such as stress, morale and teamwork. NASA has begun to collect time series psychological data on these factors by having crew members in space and analog missions fill out a survey on an almost daily basis. As of the conclusion of our project (June 1995), NASA had analyzed data from an underwater mission designed to test their data collection instrument, the IFRS (Individualized Field Recording System) survey, and was collecting data from several Antarctic traverses. The IFRS survey was scheduled to be employed on a joint Soviet-American shuttle mission. Its most likely initial use would be as a tool for helping to select crews for the international Space Station.

Our task was to design a system for incorporating eventual IFRS survey results in a model of participant behavior on long-term missions. Our goal was to implement a proof-of-concept software system to demonstrate algorithms for combining AI techniques like case-based reasoning and fuzzy logic with a statistical model of IFRS survey results and a rule-base derived from the existing literature on extreme missions.

This paper reports on our system design and its rationale. The CREW system predicts how crew members in a simulated mission would fill out their IFRS survey forms on each day of the mission, that is, how they would self-report indicators of stress, motivation, etc. As NASA collects and analyzes survey data, the CREW program can serve as a vehicle for assembling and building upon the data—entering empirical cases and tuning the rule-base. Clearly, the predictive power of CREW will depend upon the eventual quantity and quality of the survey data.

MODELING THE MISSION PROCESS

NASA is interested in how psychological factors such as those tracked in the IFRS surveys evolve over time during a projected mission's duration. For instance, it is not enough to know what the average stress level will be of crew members at the end of a nine-month mission; we need to know if any crew member will be likely to be particularly stressed at a critical point in the middle of the mission when certain actions must be taken. To obtain this level of detail of prediction, we created a *time series* model of the mission.

The model is based on standard statistical time series analysis. McDowall, et al. [6] argue for a stochastic ARIMA (Auto Regressive Integrated Moving Average) model of interrupted time series for a broad range of phenomena in the social sciences. The most general model takes into account three types of considerations: (1) trends, (2) seasonality effects and (3) interventions. An observed time series is treated as a realization of a stochastic process; the ideal model of such a process is statistically adequate (its residuals are white noise) and parsimonious (it has the fewest parameters and the greatest number of degrees of freedom among all statistically equivalent models).

(1) *Trends.* The basic model takes into account a stochastic component and three structural components. The stochastic component conveniently summarizes the multitude of factors producing the variation observed in a series which cannot be accounted for by the model. At each time t there is a stochastic component α_t which cannot be accounted for any more specifically. McDowall, et al. claim that most social science phenomena are properly modeled by first-order ARIMA models. That is, the value, Y_t of the time series at time t may be dependent on the value of the time series or of its stochastic component at time t-1, but not (directly) on the values at any earlier times. The first-order expressions for the three structural components are:

autoregressive: $Y_t = \alpha_t + \varphi Y_{t-1}$

differenced : $Y_t = \alpha_t + Y_{t-1}$

moving average : $Y_t = \alpha_t + \theta a_{t-1}$

We have combined these formulae to produce a general expression for all first-order ARIMA models:

$$Y_t = \alpha_t + \varphi Y_{t-1} + \theta \alpha_{t-1}$$

This general expression makes clear that the model can take into account trends and random walks caused by the inertia (or momentum) of the previous moment's stochastic component or by the inertia of the previous moment's actual value.

(2) *Seasonality.* Many phenomena (e.g., in economics or nature) have a cyclical character, often based on the 12 month year. It seems unlikely that such seasonality effects would be significant for NASA missions; the relevant cycles (daily and annual) would be too small or too large to be measured by IFRS time series data.

(3) *Interventions.* External events are likely to impact upon modeled time series. Their duration can be modeled as exponential decay, where the n^{th} time period after an event at time e will have a continuing impact of $Y_{e+n} = \delta^n \omega$ where $0 <= \delta <= 1$. Note that if $\delta = 0$ then there is no impact and if $\delta = 1$ then there is a permanent impact. Thus, δ is a measure of the rate of decay and ω is a measure of the intensity of the impact.

We have made some refinements to the standard time series equations, to tune them to our domain and to make them more general. First, the stochastic component, $\alpha_i(t)$, consists of a mean value, $\mu_i(t)$, and a normal distribution component governed by a standard deviation, $\sigma_i(t)$. Second, mission events often have significant effects of anticipation. In general, an event j of intensity ω_{ij} at time t_j will have a gradual onset at a rate ε_{ij} during times $t < t_j$ as well as a gradual decay at a rate δ_{ij} during times $t > t_j$. The following equation incorporates these considerations:

$$Y_{i(t)} = \alpha_{i\,(t)} + \phi_i Y_{i\,(t-1)} + \theta_i \alpha_{i\,(t-1)}$$
$$+ \sum_{\substack{j=1 \\ (for\ \ t\ <\ t_j)}}^{n} \left[\varepsilon_{ij}^{\,(t_j - t)} \omega_{ij} \right] + \sum_{\substack{j=1 \\ (for\ \ t\ \geq\ t_j)}}^{n} \left[\delta_{ij}^{\,(t - t_j)} \omega_{ij} \right]$$

where:

$Y_i(t)$ = value of factor i for a given actor in a given mission at mission time t

t_j = time of occurrence of the j^{th} of n intervening events in the mission

α = noise: a value is generated randomly with mean μ and standard deviation σ

μ = mean of noise value	$0 <= \mu <= 10$
σ = standard deviation of noise	$0 <= \sigma <= 10$
φ = momentum of value	$-1 <= \varphi <= 1$
θ = momentum of noise	$-1 <= \theta <= 1$
ε = rise rate of interruption	$0 <= v <= 1$
δ = decay rate of interruption	$0 <= \delta <= 1$
ω = intensity of interruption	$-10 <= \omega <= 10$

The model works as follows. Using IFRS survey data for a given question answered by a given crew member throughout a given mission and knowing when significant events occurred, one can use standard statistical procedures to derive the parameters of the preceding equation: μ, σ, φ and θ as well as ε, δ and ω for each event in the mission. Then, conversely, one can use these parameters to predict the results of a new proposed mission. Once one has obtained the parameters for a particular psychological factor, a crew member and each event, one can predict the values that crew member would enter for that survey question i at each time period t of the mission by calculating the equation with those parameter values.

This model allows us to enter empirical cases into a case-base by storing the parameters for each *factor* (i.e., a psychological factor for a given crew member during a given mission) or *event* (i.e., an intervention event in the given factor time series) with a description of that factor or event. To make a time series prediction of a proposed factor with its events, we retrieve a similar case, adapt it for differences from the proposed case and compute its time series values from the model equation.

USING CASE-BASED REASONING

The time series model is quite complex in terms of the number of variables and factors. It must produce different results for each time period, each kind of mission, each crew member personality, each question on the IFRS survey and each type of intervention event. To build a rule-based expert system, we would need to acquire thousands of formal rules capable of computing predictive results for all these combinations. But there are no experts on interplanetary missions who could provide such a set of rules. Nor is there data that could be analyzed to produce these rules. So we took a case-based reasoning approach. We take actual missions—including analog missions—and compute the parameters for their time series.

Each survey variable requires its own model (values for parameters μ, σ, φ, and θ), as does each kind of event (values for parameters ε, δ, and ω). Presumably, the 107 IFRS survey questions can be grouped into several factors—although this is itself an empirical question. We chose six psychological factors that we thought underlay the IFRS questionnaire: crew teamwork, physical health, mental alertness, psychological stress, psychological morale and mission effectiveness. In addition, we selected a particular question from the survey that represented each of these factors. The CREW system currently models these twelve factors.

There is no natural taxonomy of events. Our approach assumes that there are categories of events that can be modeled consistently as interventions with exponential onsets and decays at certain impact levels and decay rates.

Based on the available data, we decided to model eight event types: start of mission, end of mission, emergency, conflict, contact, illness, discovery, failure.

The case-base consists of instances of the 12 factors and the 8 event types. Each instance is characterized by its associated mission and crew member, and is annotated with its parameter values. Missions are described by 10 characteristics (variables), each rated from 0 to 10. The mission characteristics are: harshness of environment, duration of mission, risk level, complexity of activities, homogeneity of crew, time of crew together, volume of habitat, crew size, commander leadership and commander competence. Crew member characteristics are: role in crew, experience, professional status, commitment, social skills, self reliance, intensity, organization, sensitivity, gender, culture and voluntary status. In addition, events have characteristics: event type, intensity and point in mission.

Because there is only a small handful of cases of actual IFRS data available at present, additional cases are needed to test and to demonstrate the system. Approximate models of time series and interventions can be estimated based on space and analog missions reported in the literature, even if raw time series data is not available to derive the model statistically. Using these, we generate and install supplemental demo cases by perturbating the variables in these cases and adjusting the model parameters in accordance with rules of thumb gleaned from the literature on analog missions. This data base is not rigorously empirical, but it should produce plausible results during testing and demos. Of course, the database can be recreated at a later time when sufficient real data is available. At that point, NASA might change the list of factor and event types to track in the database or the set of variables to describe them. Then the actual case data would be analyzed using interrupted time series analysis to derive empirical values for μ, σ, φ, and θ for the factors.

Users of CREW enter a scenario of a proposed mission, including crew composition and mission characteristics. They also enter a series of n anticipated events at specific points in the mission period. From the scenario, the system computes values for μ, σ, φ, and θ for each behavioral factor. For events j = 1 through n, it computes values for δ_j, ε_j and ω_j. The computation of parameters is accomplished with case-based reasoning, rather than statistically. The missions or events in the case-base that most closely match the hypothesized scenario are retrieved. The parameters associated with the retrieved cases are then adjusted for differences between the proposed and retrieved cases, using rules of thumb formulated in a rule-base for this purpose. Then, using the model equation, CREW computes values of Y_t for each behavioral factor at each time slice t in the mission. These values can be graphed to present a visual image of the model's expectations for the proposed mission. Users can then modify their descriptions of the mission scenario and/or the sequence of events and re-run the analysis to test alternative mission scenarios.

CREW is basically a database system, with a system of relational files storing variable values and parameter values for historical cases and rules for case adaptation. For this reason it was developed in the FoxPro database management system, rather than in Lisp, as originally planned. FoxPro is extremely efficient at retrieving items from indexed database files, so that CREW can be scaled up to arbitrarily large case-bases with virtually no degradation in processing speed. CREW runs on Macintosh and Windows computers.

THE CASE RETRIEVAL MECHANISM

A key aspect of case-based reasoning (CBR) is its case retrieval mechanism. The first step in computing predictions for a proposed new case is to retrieve one or more similar cases from the case base. According to Schank [7], CBR adopts the dynamic memory approach of human recall.

As demonstrated in exemplary CBR systems [8], this involves a hierarchical storage and retrieval arrangement. Thus, to retrieve the case most similar to a new case, one might, for instance, follow a tree of links that begins with the mission characteristic, harshness of environment. Once one followed the link corresponding to the new case's environment, one would select the link for the next characteristic and so on until one arrived at a leaf of the tree with a particular case. The problem with this method is that not all domains can be organized in such a hierarchy meaningfully. Kolodner [9] notes that some CBR systems need to define non-hierarchical retrieval systems. In the domain of space missions, there is no clear priority of characteristics for establishing similarity of cases.

A standard non-hierarchical measure of similarity is the n-dimensional Euclidean distance, which compares two cases by adding the squares of the differences between each of the n corresponding variable values. The problem with this method is that it is intractable for large case-bases because you must compare a new case with every case in the database.

CREW adopts an approach that avoids the need to define a strict hierarchy of variables as well as the ultimately intractable inefficiency of comparing a new case to each historic case. It prioritizes which variables to compare initially in order to narrow down to the most likely neighbors using highly efficient indices on the database files. But it avoids strict requirements even at this stage.

The retrieval algorithm also responds to another problem of the space mission domain that is discussed in the section on adaptation below, the fact that there are large random variations among similar cases. This problem suggests finding several similar cases instead of just one to adapt to a new case. The case retrieval algorithm in CREW returns n nearest neighbors, where n is a small number specified by the user. Thus, parameters for new cases can be computed using adjusted values from several near neighbors, rather than just from the one nearest neighbor as is traditional in CBR. This introduces a statistical flavor to the computation in order to soften the variability likely to be present in the empirical case data.

The case retrieval mechanism consists of a procedure for finding the n most similar factors and a procedure for finding the n most similar events, given a proposed factor or event, a number n and the case-base file. These procedures in turn call various subprocedures. Each of the procedures is of computational order n, where n is the number of neighbors sought, so it will scale up with no problem for case bases of arbitrary size. Here are outlines of typical procedures:

nearest_factor(new_factor, n, file)

1. find all factor records with the same factor type, using a database index

2. of these, find the 4n with the nearest_mission

3. of these, find the n with the nearest_actor

nearest_mission(new_mission, n, file)

1. find all mission records with environment = new mission’s environment ± 1 using an index

2. if less than 20n results, then find all mission records with environment = new mission’s environment ± 2 using an index

3. if less than 20n results, then find all mission records with environment = new mission’s environment ± 3 using an index

4. of these, find the 3n records with minimal |mission’s duration - new mission’s duration| using an index

5. of these, find the n records with minimal Σdif_i^2

nearest_actor(new_actor, n, file)

1. find up to n actor records with minimal Σdif_i^2

Note that in these procedures there is a weak sense of hierarchical ordering. It is weak in that it includes only a couple of levels and usually allows values that are not exactly identical, depending on how many cases exist with identical matches. Note, too, that the n-dimensional distance approach is used (indicated by “minimal Σdif_i^2”), but only with 3*n cases, where n is the number of similar cases sought. The only operations that perform searches on significant portions of the database are those that can be accomplished using file indexes. These operations are followed by procedures that progressively narrow down the number of cases. Thereby, a balance is maintained that avoids both rigid prioritizing and intractable computations.

Case based reasoning often imposes a hierarchical priority to processing that is hidden behind the scenes. It makes case retrieval efficient without exposing the priorities to scrutiny. The preceding algorithms employ a minimum of prioritizing. In each instance, priorities are selected that make sense in the domain of extreme missions based on our understanding of the relevant literature and discussions with domain experts at NASA. Of course, as understanding of the domain evolves with increased data and experience, these priorities will have to be reviewed and adjusted.

Rules and Fuzzy Logic

Once n similar cases have been found, they must be adapted to the new case. That is, we know the time series parameters for the similar old cases and we now need to adjust them to define parameters for the new case, taking into account the differences between the old and the new cases. Because the database is relatively sparse, it is unlikely that we will retrieve cases that closely match a proposed new case. Adaptation rules play a critical role in spanning the gap between the new and the retrieved cases.

The rules have been generated by our social science team, which has reviewed much of the literature on analog missions and small group interactions under extreme conditions of isolation and confinement, e.g., [10]. They have determined what variables have positive, negligible or negative correlations with which factors. They have rated these correlations as either *strong* or *weak*. The CREW system translates the ratings into percentage correlation values. For instance, the rule, “teamwork is strongly negatively correlated with commander competence” would be encoded as a -80% correlation between the variable *commander competence* and the factor *teamwork*.

The rules function roughly as follows in CREW: one rule, for instance, is used to adjust predicted *stress* for a hypothetical mission of length *new-duration* from the stress measured in a similar mission of length *old-duration*. Suppose that the rule states that the correlation of psychological *stress* to mission *duration* is +55%. All mission factors, such as stress, are coded on a scale of 0 to 10. Suppose that the historic mission had its duration variable coded as 5 and a stress factor rating of 6, and that the hypothetical mission has a duration of 8. We use the rule to adapt the historic mission’s stress rating to the hypothetical mission given the difference in mission durations (assuming all other mission characteristics to be identical). Now, the maximum that stress could be increased and still be on the scale is 4 (from 6 to 10); the *new-duration* is greater than the old by 60% (8 - 5 = 3 of a possible 10 - 5 = 5); and the rule states that the correlation is 55%. So the predicted stress for the new case is greater than the stress for the old case by: 4 x 60% x 55% = 1.32—for a predicted stress of 6 + 1.32 = 7.32. Using this method of adapting outcome values, the values are proportional to the correlation value, to the difference between the new and old variable values and to the old outcome value, without ever exceeding the 0 to 10 range.

There are many rules needed for the system. Rules for adapting the four parameters (μ, σ, φ, and θ) of the 12 factors are needed for each of the 22 variables of the mission and actor descriptions, requiring 1056 rules. Rules for adapting the three parameters (ε, δ, and ω) of the 8 event types for each of the 12 factors are needed for each of the 24 variables of the mission, actor and intervention descriptions, requiring 6912 rules. Many of these 7968 required rules have correlations of 0, indicating that a difference in the given variable has no effect on the particular parameter.

The rules gleaned from the literature are rough descriptions of relationships, rather than precise functions. Because so many rules are applied in a typical simulation, it was essential to streamline the computations. We therefore made the simplifying assumption that all correlations were linear from zero difference between the old and new variable values to a difference of the full 10 range, with only the strength of the correlation varying from rule to rule.

However, it is sometimes the case that such rules apply more or less depending on values of other variables. For instance, the rule “teamwork is strongly negatively correlated with commander competence” might be valid only “if commander leadership is very low and the crew member’s self reliance is low”. This might capture the circumstance where a commander is weak at leading others to work on something, while the crew is reliant on him and where the commander can do everything himself. It might generally be good for a commander to be competent, but problematic under the special condition that he is a poor leader and that the crew lacks self reliance.

Note that the original rule has to do with the difference of a given variable (*commander competence*) in the old and the new cases, while the condition on the rule has to do with the absolute value of variables (*commander leadership*, *crew member’s self-reliance*) in the new case. CREW uses fuzzy logic [11] to encode the conditions. This allows the conditions to be stated in English language terms, using values like *low*, *medium*, or *high*, modifiers like *very* or *not*, and the connectives *and* or *or*. The values like *low* are defined by fuzzy set membership functions, so that if the variable is 0 it is considered completely *low*, but if it is 2 it is only partially *low*. Arbitrarily complex conditions can be defined. They compute to a numeric value between 0 and 1. This value of the condition is then multiplied by the value of the rule so that the rule is only applied to the extent that the condition exists.

The combination of many simple linear rules and occasional arbitrarily complex conditions on the rules provides a flexible yet computationally efficient system for implementing the rules found in the social science literature. The English language statements by the researchers are translated reasonably into numeric computations by streamlined versions of the fuzzy logic formalism, preserving sufficient precision considering the small effect that any given rule or condition has on the overall simulation.

THE ADAPTATION ALGORITHM

Space and analog missions exhibit large variations in survey results due to the complexity and subjectivity of the crew members' perceptions as recorded in survey forms. Even among surveys by different crew members on relatively simple missions with highly homogeneous crews, the recorded survey ratings varied remarkably. To average out these effects, CREW retrieves n nearest neighbors for any new case, rather than the unique nearest one as is traditional in CBR. The value of n is set by the user.

The parameters that model the new case are computed by taking a weighted average of the parameters of the n retrieved neighbors. The weight used in this computation is based on a similarity distance of each neighbor from the new case. The similarity distance is the sum of the squares of the differences between the new and the old values of each variable. So, if the new case and a neighbor differed only in that the new case had a mission complexity rating of 3 while the retrieved neighbor had a mission complexity rating of 6, then the neighbor's distance would be $(6-3)^2 = 9$.

The weighting actually uses a term called *importance* that is defined as (sum - distance)/(sum * (n-1)), where *distance* is the distance of the current neighbor as just defined and *sum* is the sum of the distances of the n neighbors. This weighting gives a strong preference to neighbors that are very near to the new case, while allowing all n neighbors to contribute to the adaptation process.

CONCLUSIONS AND FUTURE WORK

The domain of space missions poses a number of difficulties for the creation of an expert system:

Too little is known to generalize formal rules for a rule-based system.

A model of the temporal mission process is needed more than just a prediction of final outcomes.

The descriptive variables cannot be put into a rigid hierarchy to facilitate case-based retrieval.

The case-base is too sparse and too variable for reliable adaptation from one nearest neighbor case.

The rules that can be gleaned from available data or relevant literature are imprecise.

Therefore, we have constructed a hybrid system that departs in several ways from traditional rule-based as well as classic case-based systems. CREW creates a time series model of a mission, retrieving and adapting the parameters of the model from a case base. The retrieval uses a multi-stage algorithm to maintain both flexibility and computational tractability. An extensive set of adaptation rules overcomes the sparseness of the case base, with the results of several nearest neighbors averaged together to avoid the unreliability of individual cases.

Our proof-of-concept system demonstrates the tractability of our approach. For testing purposes, CREW was loaded with descriptions of 50 hypothetical missions involving 62 actors. This involved 198 intervention parameters, 425 factor parameters and 4,047 event parameters. Based on our reading of the relevant literature, 7,968 case adaptation rule correlation figures were entered. A number of fuzzy logic conditions were also included for the test cases. Given a description of a crew member and a mission, the CREW system predicts a series of one hundred values of a selected psychological factor in a minute or two on a standard Macintosh or Windows desktop computer.

Future work includes expanding the fuzzy logic language syntax to handle more subtle rules. Our impression from conflicting conclusions within the literature is that it is unlikely that many correlation rules hold uniformly across entire ranges of their factors.

We would also like to enhance the explanatory narrative provided by CREW in order to increase its value as a research assistant. We envision our system serving as a tool to help domain experts select astronaut crews, rather than as an automated decision maker. People will want to be able to see and evaluate the program's rationale for its predictions. This would minimally involve displaying the original sources of cases and rules used by the algorithms. The most important factors should be highlighted. In situations strongly influenced by case adaptation rules or fuzzy logic conditions derived from the literature, it would be helpful to display references to the sources of the rules if not the relevant excerpted text itself.

Currently, each crew member is modeled independently; it is undoubtedly important to take into account interactions among them as well. While crew interactions indirectly affect survey results of individual members (especially to questions like: How well do you think the crew is working together today?), additional data would be needed to model interactions directly. Two possible approaches suggest themselves: treating crew interaction as a special category of event or subjecting data from crew members on a mission together to statistical analyses to see how their moods, etc. affect one another. Taking interactions into account would significantly complicate the system and would require data that is not currently systematically collected.

Use of the system by NASA personnel will suggest changes in the variables tracked and their relative priority in the processing algorithms; this will make end-user modifiability facilities desirable. In order to quickly develop a proof-of-concept system, we hard-coded many of the algorithms described in this paper. However, some of these algorithms make assumptions about, for instance, what are the most important factors to sort on first. As the eventual system users gain deeper understanding of mission dynamics, they will want to be able to modify these algorithms. Future system development should make that process easier and less fragile.

Data about individual astronauts, about group interactions and about mission progress at a detailed level is not public information. For a number of personal and institutional reasons, such information is closely guarded. Combined with the fact that NASA was just starting to collect the kind of time series data that CREW is based on, that made it impossible for us to use empirical data in our case base. Instead, we incorporated the format of the IFRS surveys and generated plausible data based on the statistical results of completed IFRS surveys and the public literature on space and analog missions. When NASA has collected enough empirical cases to substitute for our test data, they will have to enter the new parameters, review the rule base, and reconsider some of the priorities embedded in our algorithms based on their new understanding of mission dynamics. However, they should be able to do this within the computational framework we have developed, confident that such a system is feasible. As NASA collects more time series data, the CREW database will grow and become increasingly plausible as a predictive tool that can assist in the planning of expensive and risky interplanetary missions.

REFERENCES

[1] American Astronomical Society (1966) *Stepping Stones to Mars Meeting*. Amer. Inst. of Aero. & Astro. New York.

[2] McKay, C (ed.) (1985) *The Case for Mars II.* Am. Astro. Society. San Diego, CA.

[3] Harrison, A, Clearwater, Y, McKay C (1991) *From Antarctica to Outer Space: Life in Isolation and Confinement.* Behavioral Science. Vol. 34, pp. 253-271.

[4] Collins, D (1985) Psychological Issues Relevant to Astronaut Selection for Long Duration Spaceflight: A Review of the Literature. USAF Systems Command. Brooks Air Force Base. Texas.

[5] Owen, R, Holland, A, Wood, J (1993) A Prototype Case-Based Reasoning Human Assistant for Space Crew Assessment and Mission Management. In Fayyad, U, Uthurusamy, F (eds.) (1993) *Applications of Artificial Intelligence 1993: Knowledge-Based Systems in Aerospace and Industry.* SPIE vol. 1963, pp. 262-273. Bellingham, WA.

[6] McDowall, D, McLeary, R, Meidinger, E, Hay, R Jr. (1980) *Interrupted Time Series Analysis*. Sage. Beverly Hills, CA.

[7] Schank, R (1982) *Dynamic Memory*. Cambridge Univ. Press. Cambridge.

[8] Riesbeck, C & Schank, R (1989) *Inside Case-Based Reasoning*. Lawrence Earlbaum. Hillsdale, NJ.

[9] Kolodner, J (1993) *Case-Based Reasoning*. Morgan Kaufmann. San Mateo, CA.

[10] Radloff, R, Helmreich, R (1968) *Groups Under Stress: Psychological Research in Sealab II.* Appleton Century Crofts. New York.

[11] Cox, E (1994) *The Fuzzy Systems Handbook*. Academic Press. Boston.

SHARE GLOBALLY, ADAPT LOCALLY: SOFTWARE ASSISTANCE TO LOCATE AND TAILOR CURRICULUM POSTED TO THE INTERNET

ABSTRACT

Many teachers yearn to break through the confines of traditional textbook-centered teaching to present activities that encourage students to explore and construct their own knowledge. But this *requires developing innovative materials and curriculum* tailored to local students. Teachers have neither the time nor the information to do much of this from scratch.

The Internet provides a medium for sharing innovative educational resources globally. School districts and teacher organizations have already begun to post curriculum ideas on Internet servers. However, just storing unrelated educational materials on the Internet does not by itself solve the problem. It is too hard to find the right resources to meet specific needs. Teachers need productivity software for *locating* sites of materials across the network, *searching* the individual curriculum sources, *adapting* retrieved materials to their classrooms, *organizing* these resources in coherent lesson plans, and *sharing* their experiences across the Internet.

We have designed and prototyped a Teacher's Curriculum Assistant (TCA) that provides *software support for teachers to make effective use of educational resources posted to the Internet*. TCA maintains information for finding educational resources distributed on the Internet. It provides query and browsing mechanisms for exploring what is available. Tools are included for tailoring retrieved resources, creating supplementary materials, and designing innovative curriculum. TCA encourages teachers to annotate and upload successfully used curriculum to Internet servers to share their ideas with other teachers. In this paper we motivate the need for such computer support and discuss what we have learned from designing TCA.

INTRODUCTION

The Internet has the potential to transform educational curriculum development beyond the horizons of our foresight. The process has begun, as educators across the country start to post their favorite curriculum ideas for others to share. Already, this first tentative step has revealed the difficulties inherent in using such potentially enormous, loosely structured sources of information. Teachers wandering around the Internet looking for ideas to use in their classrooms confront a set of problems that will not go away by itself as the Internet becomes a more popular medium for sharing curriculum—on the contrary:

1. Teachers have to *locate* sites of curriculum ideas scattered across the network; there is currently no system for announcing the locations of these sites.
2. They have to *search* through the offerings at each site for useful items. While some sites provide search mechanisms for their databases, each has different interfaces, tools, and indexing schemes that must be learned before the curricula can be accessed.
3. They have to *adapt* items they find to the needs of their particular classroom: local standards, the current curriculum, their own teaching preferences, and the needs or learning styles of their various students.
4. They have to *organize* the new ideas in coherent curricula that build toward long-term pedagogical goals.
5. They have to *share* their experiences using the curriculum or their own new ideas with others who use the resources.

In many fields, professionals have turned to *productivity software* to help them manage such tasks involving complex sources of information. We believe that teachers should be given similar computer-based tools to meet the problems listed above. If this software is designed to empower teachers—perhaps in conjunction with their students—in open-ended ways, opportunities will materialize that we cannot now imagine.

In this article, we consider how the sharing of curriculum ideas over the Internet can be made more effective in transforming education. We motivate specific issues in the design of productivity software for curriculum development by classroom teachers, and introduce the Teacher's Curriculum Assistant (TCA) we are building for this purpose. First, we discuss the nature of constructivist curriculum, contrasting it with traditional approaches based on behaviorist theory. Then we present an example of a problem-solving environment for high school mathematics students. The example illustrates why teachers need help to construct this kind of student-centered curriculum. We provide a scenario of a teacher developing curriculum using productivity software like TCA, and conclude by discussing some issues we feel will be important in *maximizing the effectiveness of the Internet* as a medium for the dissemination of innovative curriculum for educational reform.

THE PROBLEM OF CURRICULUM IN EDUCATIONAL REFORM

The distribution of curriculum over the Internet and the use of productivity software for searching and adapting posted ideas could benefit any pedagogical approach. However, it is particularly crucial for advancing *reform* in education.

The barriers to educational reform are legion, as many people since John Dewey have found. Teachers, administrators, parents, and students must all be convinced that traditional schooling is not the most effective way to provide an adequate foundation for life in the future. They must be trained in the new sensitivities required. Once everyone agrees and is ready to implement the new approach there is still a problem: what activities and materials should be presented on a day to day basis? This concrete question is the one that Internet sharing can best address. We generalize the term *curriculum* to cover this question.

Consider curriculum for mathematics. Here, the reform approach is to emphasize the qualitative understanding of mathematical ways of thinking, rather than to stress rote memorization of quantitative facts or "number skills". *Behaviorist* learning theory supported the view that one method of training could work for all students; reformers face a much more complex challenge. There is a growing consensus among educational theorists that different students in different situations construct their understandings in different ways [1]. This approach is often called *constructivism* or constructionism [2]. It implies that teachers must creatively structure the learning environments of their students to provide opportunities for discovery and must guide the individual learners to reach insights in their own ways.

Behaviorism and constructivism differ primarily in their views of how students build up their knowledge. Traditional, rationalist education assumed that there was a logical sequence of facts and standard skills that had to be learned successively. The problem was simply to transfer bits of information to students in a logical order, with little concern for how students acquire knowledge. Early attempts at designing educational software took this approach to its extreme, breaking down curriculum into isolated atomic propositions and feeding these predigested facts to the students. This approach to education was suited to the industrial age, in which workers on assembly lines performed well-defined, sequential tasks.

According to constructivism, learners *interpret* problems in their environments using *conceptual frameworks* that they developed in the past [3]. In challenging cases, problems can require changes in the frameworks. Such conceptual change is the essence of learning: one's understanding evolves in order to comprehend one's environment [4]. To teach a student a mathematical method or a scientific theory is not to place a set of propositional facts into her mind, but to give her a new tool that she can make her own and use in her own ways in comprehending her world.

Constructivism does not entail the rejection of curriculum. Rather, it requires a more complex and flexible curriculum. Traditionally, curriculum consisted of a textual theoretical lesson, a set of drills for students to practice, and a test to evaluate if the students could perform the desired behaviors. In contrast, *a constructivist curriculum* might target certain cognitive skills, provide a setting of resources and activities to serve as a catalyst for the development of these skills, and then offer opportunities for students to articulate their evolving understandings [5]. The cognitive skills in math might include qualitative reasoning about graphs, number lines, algorithms, or proofs, for example.

We believe that the movement from viewing curriculum as fact-centered to viewing it as cognitive-tool-centered is appropriate for the post-modern (post-industrial, post-rationalist, post-behaviorist) period. Cognitive tools include, importantly, *alternative knowledge representations* [6]. As researchers in artificial intelligence, we know that knowledge representations are key to characterizing or modeling cognition. We have also found that professionals working in typical contemporary occupations focus much of their effort on developing and using alternative knowledge representations that are adapted to their tasks [7]. Curricula to prepare people for the next generation of jobs would do well to familiarize students with the creation and use of alternative conceptual representations.

A DIVERSE LEARNING ECOLOGY

We are interested in helping teachers to create learning environments that stimulate the construction and evolution of understanding through student exploration using multiple conceptual representations. *A stimulating learning environment is one with a rich ecology, in which many elements interact in subtle ways.* In this section we present an illustration of a rich ecology for learning mathematical thinking that includes: inductive reasoning, recursive computation, spreadsheet representation, graphing, linear equations, and programming languages.

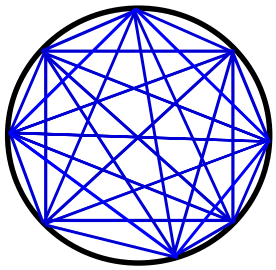


Figure 1. Regions of a circle; n = 8..

A typical curriculum suggestion that might be posted on an educational resources listing on the Internet is the *problem of regions of a circle*: Given n points on the circumference of a circle, what is the maximum number of regions you can divide the circle into by drawing straight lines connecting the points? (See Figure 1.) For instance, connecting two points divides the circle into two regions; connecting three points with three lines creates four regions. This is a potentially fascinating problem because its subtleties can be explored at length using just algebra and several varieties of clear thinking.

The problem with this curriculum offering as an Internet posting is that it has not been placed in a rich setting. To be useful, a fuller curriculum providing a set of conceptual tools is needed. For instance, a discussion of inductive reasoning brings out some of the character of this particular problem. If one counts the number of regions, R(n), for n = 1 to 6, one obtains the doubling series: 1, 2, 4, 8, 16, 31. Almost! One expects the last of these numbers to be 32, but that last region is nowhere to be found. For larger n, the series diverges completely from the powers of 2. Why? Here *inductive reasoning* can come to the rescue of the hasty inductive assumption—if, that is, the problem is accompanied by a discussion of inductive reasoning.

Consider the general case of n points. Assume that you know the answer for n-1 points and think about how many new regions are created by adding the n-th point and connecting it to each of the n-1 old points. There is a definite pattern at work here. It may take a couple days of careful thought to work it out. It would also help if the *sigma notation* for sums of indexed terms is explained as a tool for working on the problem. Perhaps a group effort will be needed to check each step and avoid mistakes.

At this point, a teacher might introduce the notion of *recursion* and relate it to induction. If the students can *program in Logo or Pascal* (programming languages that can represent recursive processes), they could put the general formula into a simple but powerful program that could generate results for hundreds of values of n very quickly without the tedious and error-prone process of counting regions in drawings. It would be nice to formalize the derivation of this result with a *deductive proof*, if the method of formulating proofs has been explained.

Now that students are confident that they have the correct values for many n, they can enter these values in a *spreadsheet* to explore them. The first representation they might want to see is a *graph* of R(n) *vs*. n. On the spreadsheet they could make a column that displays the difference between each R(n) and its corresponding R(n-1). Copying this column several times, they would find that the fourth column of differences is constant. This result means that R(n) follows a fourth order equation, that can be found by solving *simultaneous linear equations*.

The point of this example is that sharing the isolated statement of the problem is not enough. The rich learning experience involves being introduced to alternative representations of the problem: induction, recursion, spreadsheet differences, graphs, computer languages, simultaneous equations, etc. There is not one correct method for tackling a problem like this; a mathematically literate person needs to be able to view the problem's many facets through several conceptual frameworks.

Curriculum in the new paradigm typically consists of stimulating problems immersed in environments with richly interacting ecologies, including: cognitive skills, knowledge representations, computational tools, related problems, and reference materials. Perhaps a creative teacher with unlimited preparation time could put these materials together. However, the reality is that teachers deserve all the support they can get if they are to prepare and present the complex learning ecologies that constructivist reforms call for. Computer support for curriculum development should make the kinds of resources shown in Figure 2 readily available.

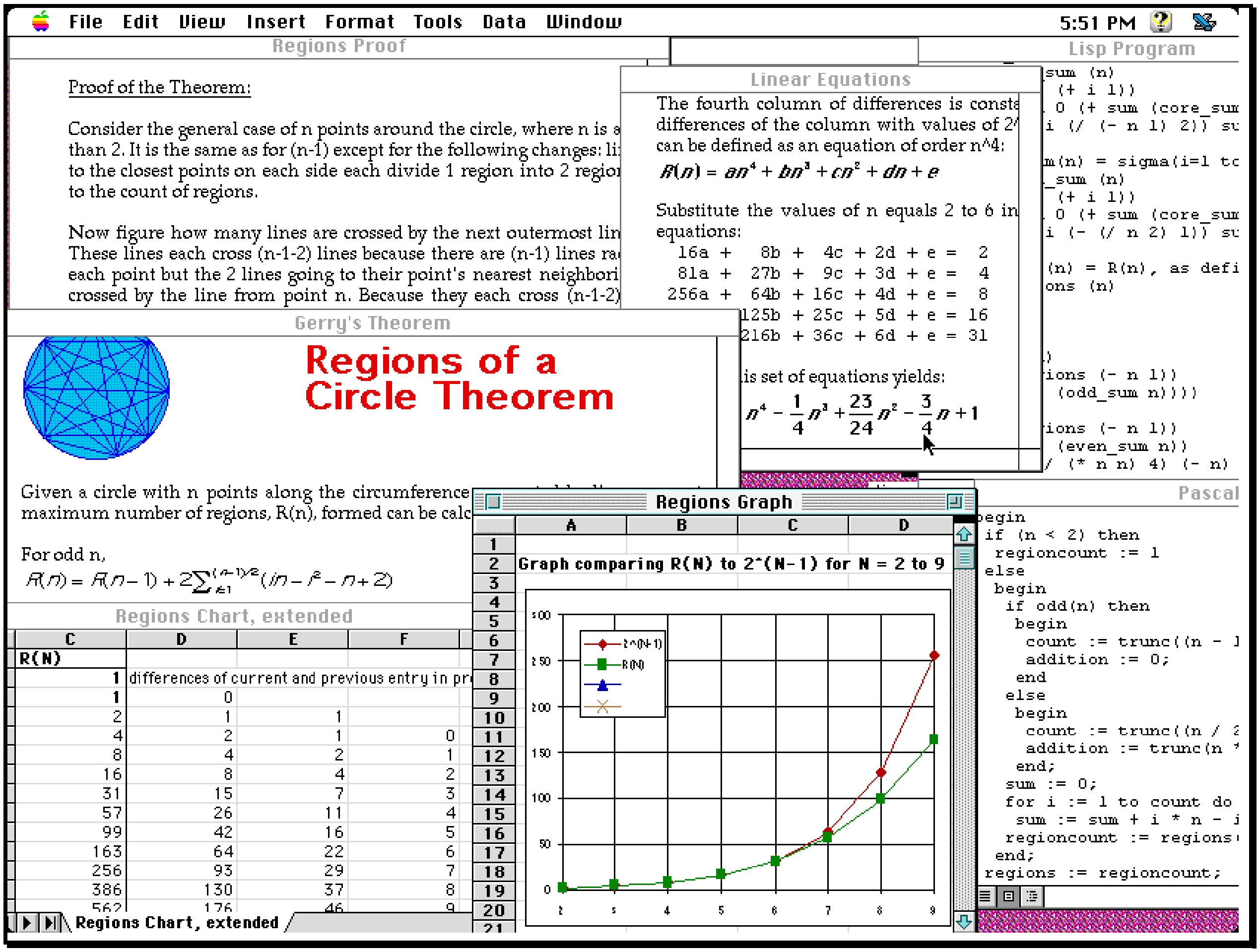


Figure 2. A number of multimedia resources related to the "regions of a circle" problem. These include textual documents, drawings, equations, spreadsheets, graphs, and computer program source code.

From Database to Design Environment

Curriculum planning for learning ecologies is not a simple matter of picking consecutive pages out of a standard textbook or of working out a sequential presentation of material that builds up to fixed learning achievements. Rather, it is a matter of *design*. To support teachers in developing curriculum that achieves this, we must go beyond databases of isolated resources to provide *design environments for curriculum development*.

It may seem to be an overwhelming task to design an effective learning environment for promoting the development of basic cognitive skills. However, dozens of reform curricula have already been created. *The problem now is to disseminate these in ways that allow teachers to adapt them* to their local needs and to reuse them as templates for additional new curricula. It is instructive to look at a recent attempt to make this curriculum available. The "MathFinder CD-ROM: a collection of resources for mathematics reform" excerpts materials from thirty new math curricula [8]. Like the posting of curriculum ideas at several Internet sites, this is an important early step at electronic dissemination.

Unfortunately, MathFinder has a number of serious limitations due to its CD-ROM (read-only) format. It relies on a fixed database of resources that allows resources to be *located* but not expanded or revised. Its indexing is relatively simple—primarily oriented toward illustrating a particular set of math standards—yet its *search mechanism is cumbersome* for many teachers. Since its resources are stored in bitmap images, they *cannot be adapted* in any way by teachers or students. Moreover, MathFinder provides *no facility for organizing resources into curricula*—despite the fact that most of the resources it includes are excerpted from carefully constructed curricula. Because it is sold as a read-only commodity, MathFinder *does not allow teachers to share* their experiences with annotations or to add their own curricular ideas. Thus, of the five issues listed in the Introduction, MathFinder only provides a partial solution to the issues of location and search.

An alternative approach is suggested by our work on *domain-oriented design environments* [9-13]. A software design environment provides a flexible workspace for the construction of artifacts and places useful design tools and materials close at hand. A design environment for curriculum development goes substantially beyond a database of individual resources. We have built a prototype version of a Teacher's Curriculum Assistant (TCA) based on this approach. TCA includes a *catalog* of previously designed curricula that can be reused and modified. It has a *gallery* of educational resources that can be inserted into partial curriculum designs. There is a *workspace*, into which curricula from the catalog can be loaded and resources from the gallery inserted. It is also possible for a teacher to specify criteria for the desired curriculum. The *specifications* are used for searching the case-base of curriculum, adapting the resources, and *critiquing* new designs.

TCA allows teachers to download curricular resources from the Internet and to create coherent classroom activities tailored to local circumstances. In particular, TCA addresses the set of problems identified in the Introduction:

1. TCA is built on a database of information about educational resources posted to the Internet, so it provides a mechanism for teachers to *locate* sources of curriculum ideas at scattered Internet sites.
2. The TCA database indexes each resource in a uniform way, allowing teachers to *search* for all items meeting desired conditions.
3. TCA includes tools to help teachers *adapt* items they find to the needs of their classroom.
4. TCA provides a design workspace for *organizing* retrieved ideas into lesson plans that build toward long-term goals.
5. TCA lets teachers conveniently *share* their experiences back through the Internet.

To illustrate how TCA works, each of these points will be discussed in the following sections. These sections present a scenario of a teacher using TCA to locate resources, search through them, adapt selected resources, organize them into curriculum, and share the results with other teachers.

SCENARIO STEP 1: LOCATING CURRICULUM

Assume that you are a high school mathematics teacher using TCA. In the coming year you have to introduce some geometric concepts like Pythagoras' Theorem and deductive proofs. More generally, you might like to discuss the ubiquity of patterns and ways to represent them mathematically. The TCA Find menu lets you search for semester themes and their constituent weekly units and lesson plans related to these topics. TCA distinguishes four levels of curriculum available on the Internet:

- A *theme* is a major curriculum, possibly covering a semester or a year of school and optionally integrating several subjects. A theme consists of multiple teaching units.
- A weekly *unit* is part of a theme, typically one week of lessons for a single subject. A unit is described by its constituent daily lesson plans.
- A *plan* is one day's lesson for a class. A lesson plan might include a number of resources, such as a lecture, a reading, an exercise or project, perhaps a quiz, and a homework assignment.
- A *resource* is an element of a lesson plan. It might be a text, available as a word processing document. It could also be a video clip, a spreadsheet worksheet, a graphic design, or a software simulation. Resources are the smallest units of curriculum indexed by TCA.

TCA lets you locate relevant curriculum by analyzing information stored on your computer about items available on the Internet. Along with the TCA software on your computer there is a case-base of summaries (indexes) of curriculum and resources that can be downloaded. These summary records reference curriculum and resources that have been posted to Internet nodes around the world. In addition to containing the Internet address information needed for downloading an item, a record contains a description of the item, so that you can decide whether or not it is of interest.

After you have selected a set of interesting items based on the information in the case-base, TCA downloads the items to your computer. This happens without you having to know where they were located or how to download them. The items are then available for modification, printing, or distribution to your students. If Internet traffic is slow, you may opt to download batches of curriculum and resources over night and then work with them the next day.

SCENARIO STEP 2: SEARCHING FOR RESOURCES

TCA provides a combination of query and browsing mechanisms to help you select curriculum of interest and to find resources that go with it. You can start by specifying that you want curriculum for tenth grade mathematics. Then you can browse through a list of themes that meet the specification. If the list is too long, narrow down your search criteria.

The *theme* named "A Look at the Greek Mind" is summarized as: "This is an integrated curriculum that explores myth, patterns and abstract reasoning." It emphasizes patterns and is likely to include Pythagoras' theorem. Click on this theme in the list. Your computer now displays summaries of the *units* that make up the curriculum for that theme. This list shows three weekly units. Select the week described as "Abstract thinking: number theory and deductive reasoning."

You now see summaries of that week's five daily *lesson plans*. Look at the geometry example for day 3, "Inductive reasoning example: regions of a circle." Select that one and the screen changes to show the lesson plan in Figure 3. It lists all the *resources* suggested for that period: two lecture topics, a class exercise, three activities for small groups and a homework assignment.

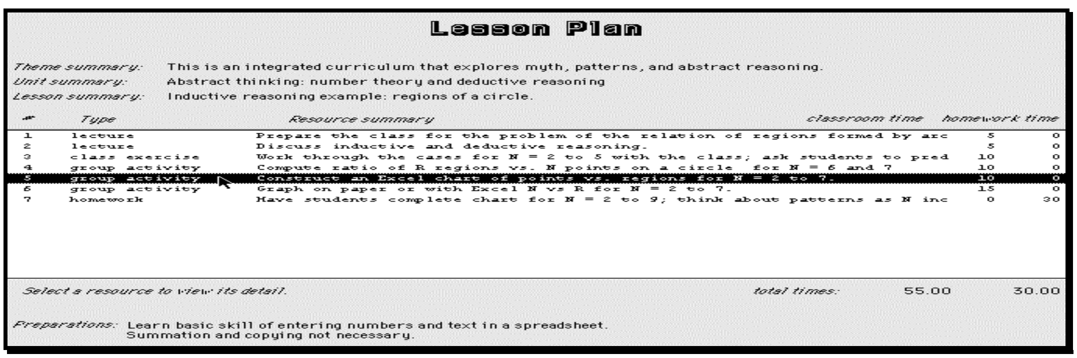


Figure 3. Screen image of the lesson plan workspace. A number of resources (lectures, exercises, group activities, and homework) related to the regions of a circle problem are assembled for a day's class. Note that total class time and homework time are computed and teacher preparations for the resources are listed below the workspace.

Notice resource #5 where students create a spreadsheet chart: "Group activity: Construct an Excel chart of points vs. regions for N= 2 to 7." Select it by clicking the mouse on the summary of that resource. Figure 4 shows the detail for that resource, including its index values.

The description contained in the case-base for each posted resource is organized as a set of 24 indexes and annotations, such as: recommended grade level, content area, pedagogical goal, instructional mode, prerequisites, materials used, required time, and the like. TCA includes search mechanisms that allow you to specify your curriculum needs using combinations of these indexes. Resources are also cross-referenced so that you can retrieve many different resources that are related to a given one. Thus, once you have found the "problem of regions of a circle", you can easily locate discussions of inductive reasoning, formal proofs, recursion, simultaneous linear equations, sample programs in Logo or Pascal, spreadsheet templates for analyzing successive differences, and graphing tools. You can also find week-long units that build on geometric problems like this one, with variations for students with different backgrounds, learning styles, or interests. TCA allows you to search both top-down from themes to resources and bottom-up from resources to curriculum.

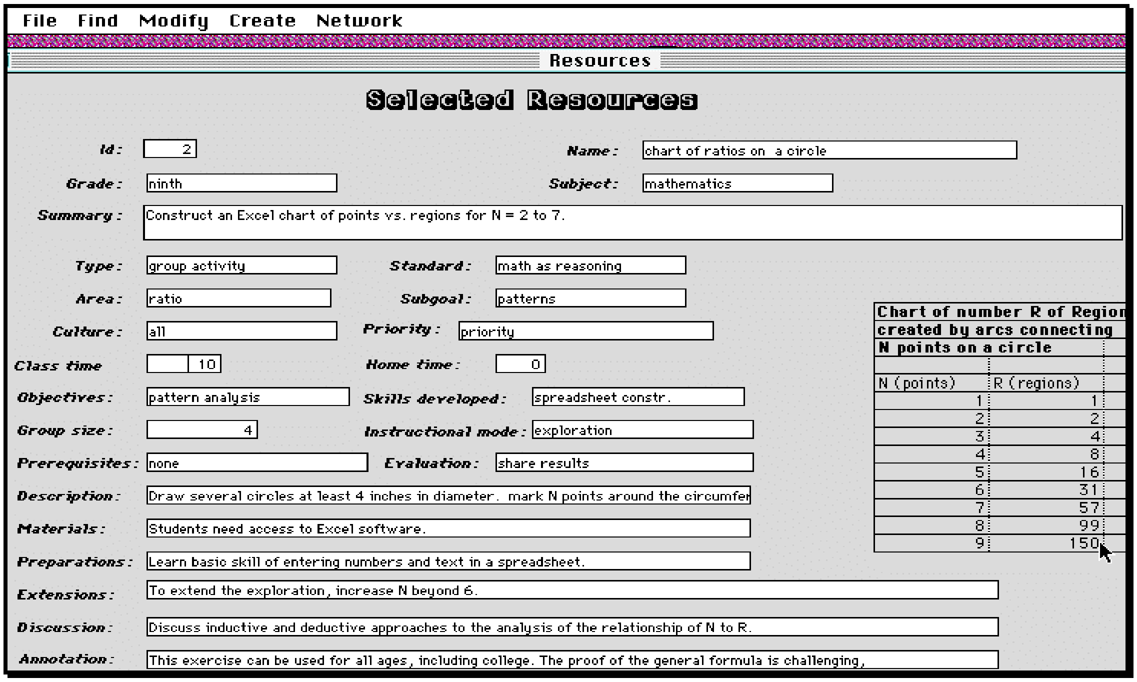


Figure 4. Screen image of a TCA display of the indexing for a resource. The resource is a spreadsheet, which is also shown in the screen.

SCENARIO STEP 3: ADAPTING TO LOCAL NEEDS

Adaptation tools are available in TCA for resources that have been downloaded from the Internet. The TCA system can often *make automated suggestions* for adapting a resource to the specification given in the search process. For instance, if you retrieve a resource that was targeted for 11th grade when you are looking for 10th grade material, then TCA might suggest allowing your students more time to do the tasks or might provide more supporting and explanatory materials for them. In general, you will need to make the adaptations; even where the software comes up with suggestions, you must use your judgment to make the final decision.

While TCA can automate some adaptation, most tailoring of curriculum requires hands-on control by experienced teachers. Sometimes TCA can support your efforts by *displaying useful information*. For instance, if you are adapting resources organized by national standards to local standards you might like your computer to display both sets of standards and to associate each local standard with corresponding national standards. In other situations, perhaps involving students whose first language is not English, TCA might link a resource requiring a high level of language understanding to a supplementary visual presentation.

The adaptation process relies on alternative *versions* of individual resources being posted. TCA helps you adjust to different student groups, teaching methods, and time constraints by retrieving alternative versions of resources that provide different motivations, use different formats, or go into more depth. You can substitute these alternative resources into lesson plans; they can then be modified with multimedia editing software from within TCA.

Included in Figure 4 was a reduced image of the spreadsheet itself. If you click on this image, TCA brings up the commercial software application in which the document was produced. So you can now *edit and modify* the copy of this document which appears on your screen. You need not leave TCA to do this. Then you can print out your revised version for your students or distribute it directly to their computers. In this way, you can use your own ideas or those of your students to modify and enhance curricular units found on the Internet.

Just as it is important for teachers to adapt curriculum to their needs, it is desirable to have resources that students can tailor. Current software technology makes this possible, as illustrated by a number of simulations in the Exploratorium described in this issue [14].

SCENARIO STEP 4: ORGANIZING RESOURCES INTO LESSON PLANS

The lesson plan is a popular representation for curriculum. It provides a system for organizing classroom activities. TCA uses the *lesson plan metaphor* as the basis for its design workspace. You can start your planning by looking at downloaded lesson plans and then modifying them to meet your local needs.

The TCA workspace for designing lesson plans was shown in Figure 3. In addition to summaries of each resource, the workspace lists the time required by each resource, both in class and at home. These times are totaled at the bottom of the list. This provides an indication of whether there is too much or too little instructional material to fill the period. You can then decide to add or eliminate resources, or adjust their time allowances. The total homework time can be compared to local requirements concerning homework amounts.

TCA incorporates computational *critics* [11, 12]. Critics are software rules that monitor the curriculum being constructed and verify that specified conditions are maintained. For instance, critics might inform you if the time required for a one-day curriculum exceeds or falls short of the time available.

Scenario step 5: sharing new experiences

Once you have developed curricula and used them successfully in the classroom, you may want to share your creations with other teachers. This way*, the pool of ideas on the Internet will grow and mature*. TCA has facilities for you to annotate individual resources and curricular units at all levels with descriptions of how they worked in your classroom. This is part of the indexing of the resource or unit.

Assume that you downloaded and used the “regions of a circle” resource and modified it based on your classroom experience. Now you want to upload your version back to the Internet. TCA automates that process, posting the new resource to an available server and adding the indexes for it to the server used for distributing new indexes. Because the indexing of your revision would be similar to that of the original version of the resource, other teachers looking at the “regions of a circle” resource would also find your version with your comments. In this way, the Internet pool of resources serves as a medium of communication among teachers about the specific resources. It is in such ways that we hope the use of the Internet for curriculum development will go far beyond today’s first steps.

WHAT WE HAVE LEARNED

We conceptualize the understanding we have reached through our work on TCA in five principles:

1. Most resources should be *located* at distributed sites across the Internet, but carefully structured summaries (indexes) of them should be maintained on teachers’ local computers.
2. The *search* process should be supported through a combination of query and browsing tools that help teachers explore what is available.
3. *Adaptation* of tools and resources to teachers and students is critical for developing and benefiting from constructivist curriculum.
4. Resources must be *organized* into carefully designed curriculum units to provide effective learning environments.
5. The Internet should become a medium for *sharing* curriculum ideas, not just accessing them.

We have designed and prototyped a system to assist teachers in developing curriculum for educational reform. We must now refine all aspects of the system by working further with classroom teachers and curriculum developers. While the approach of TCA appeals to teachers who have participated in its design, its implementation must still be tuned to the realities of the classroom.

The distribution of resources and indexes prototyped in TCA has attractive advantages. Because the actual multimedia resources (text, pictures, video clips, spreadsheet templates, HyperCard stacks, software applications) are distributed across the Internet, there is no limit to the quantity or size of these resources and no need for teachers to have large computers. Resources can be posted on network servers maintained by school districts, regional educational organizations, textbook manufacturers, and other agencies. Then the originating agency can maintain and revise the resources as necessary.

However, the approach we advocate faces a major institutional challenge: the standardization of resource indexing. The difficulty with this approach is the need to index every resource and to distribute these indexes to every computer that runs TCA. This involves (a) implementing a distribution and updating system for the case-base index records and (b) establishing the TCA indexing scheme as a standard.

The distribution and updating of indexes can be handled by tools within TCA and support software for major curriculum contributors. However, the standardization requires coordination among interested parties. Before any teachers can use TCA there must be useful indexed resources available on the network, with comprehensive suggested lesson plans. We hope to initiate cooperation among federally-funded curriculum development efforts, textbook publishers, software publishers, and school districts. If successful, this will establish a critical mass of curriculum on the Internet accessible by TCA. Then the Internet can begin to be an effective medium for the global sharing of locally adaptable curriculum.

ACKNOWLEDGMENTS

This paper describes work done at Owen Research with support by DOE grant DE-FG03-93ER81588 and NSF grant III-9360544. We wish to acknowledge encouragement from Len Scrogan, Technology Specialist in the Curriculum and Instruction Division of Boulder Valley Public Schools, and Jim Spohrer of Apple Computers. Our design environment approach grows out of research at the Center for LifeLong Learning and Design, University of Colorado.

REFERENCES

1. Greeno, J. For research to reform education and cognitive science. In: Penner, L., Batsche, G., Knoff, H., Nelson, D. *The Challenge in mathematics and science education: Psychology's response.* APA Press. 1993.
2. Papert, S. The children's machine: Rethinking school in the age of the computer. Basic Books. 1993.
3. Roschelle, J. Learning by collaborating: Convergent conceptual change. *Journal of the learning sciences*, 2, 235-276, 1992.
4. Stahl, G. Supporting situated interpretation. *Proceedings of the cognitive science society*, 965-970, 1993.
5. National Council of Teachers of Mathematics. Curriculum and evaluation standards for school mathematics. 1989.
6. Norman, D. *Things that make us smart*. Addison Wesley. 1993.
7. Sumner, T. The high-tech toolbelt: A study of designers in the workplace. To appear in: *Proceedings of the ACM CHI Conference*. 1995.
8. Kreindler, L., Zahm, B. MathFinder sourcebook: A collection of resources for mathematics reform. The Learning Team. 1992.
9. Stahl, G., McCall, R., Peper, G. Extending hypermedia with an inference language: An alternative to rule-based expert systems. *Proceedings of the IBM ITL conference: Expert systems*, 160-167, 1992.
10. Stahl, G. *Interpretation in design: The problem of tacit and explicit understanding in computer support of cooperative design*. Ph.D. dissertation. UMI #9423544. Department of Computer Science. University of Colorado at Boulder. Technical Report CU-CS-688-93. 1993.
11. Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., Sumner, T. Embedding critics in design environments. *The knowledge engineering review*, 8, 157–164, 1993.
12. Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., Sumner, T. Embedding computer-based critics in the contexts of design. *Proceedings of the ACM CHI Conference*, 157–164, 1993.
13. Repenning, A., Sumner, T. Agentsheets: A medium for creating domain-oriented visual programming languages. To appear in: *IEEE Computer*. Special issue on visual programming. March, 1995.
14. Ambach, P., Perrone, C., Repenning, A. Remote exploratoriums: Combining networking media and design environments to support engaged learning. *Computers & Education*, this issue, 1995.

COLLABORATIVE INFORMATION ENVIRONMENTS TO SUPPORT KNOWLEDGE CONSTRUCTION BY COMMUNITIES

ABSTRACT

In the information age, lifelong learning and collaboration are essential aspects of most innovative work. Fortunately, the computer technology which drives the information explosion also has the potential to help individuals and groups to learn much of what they need to know on demand. 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 paper 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.

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, 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, 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 (Stahl, 1998).

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 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 1 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 2 sketches a theory of how software productivity environments for design work by individuals can be extended to support organizational learning in collaborative work settings known as communities of practice; a scenario of a prototype system called WEBNET illustrates this. Section 3 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.

SECTION 1. AUGMENTING THE WORK OF INDIVIDUAL DESIGNERS

In this first 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 & Redmiles, 1998).

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 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 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 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).

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 Figure 1). NETSUITE contains a wealth of domain knowledge. Its palette of devices that can be placed in the construction area numbers over 5,000, 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.

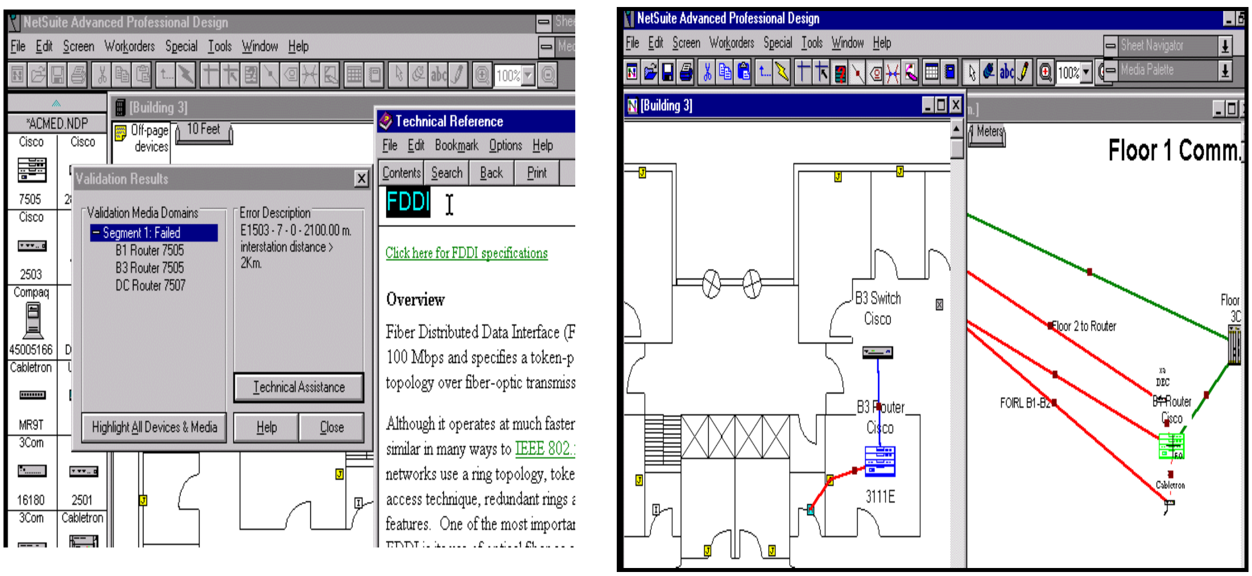


Figure 1. Two views of NETSUITE. In the left view, the system has noted that a cable length specification for a FDDI network has been exceeded in the design and the system has delivered information about the specification and affected devices. In the right view, parts of the network viewed in physical 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.

The Need to Go Further

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. A 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 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:

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 organizational 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 organization 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.

SECTION 2. 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.

Community Memories

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; Giddens, 1984; 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, 1986; Yates & 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 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 paper. For a review of common issues and positive and negative examples of responses, see (Grudin, 1990; Orlikowski, 1992; Orlikowski et al., 1995).)

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, 1984; Lave & 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 organizational learning possible within the group (Ackerman & McDonald, 1996; Argyris & Schön, 1978; Borghoff & Parechi, 1998; Buckingham Shum & Hammond, 1994; Senge, 1990).

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 the 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 like those to be discussed in Section 3 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 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 to or want to, and can communicate about it. This illustrates computer support for collaborative learning with digital memories belonging to communities of practice.

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.

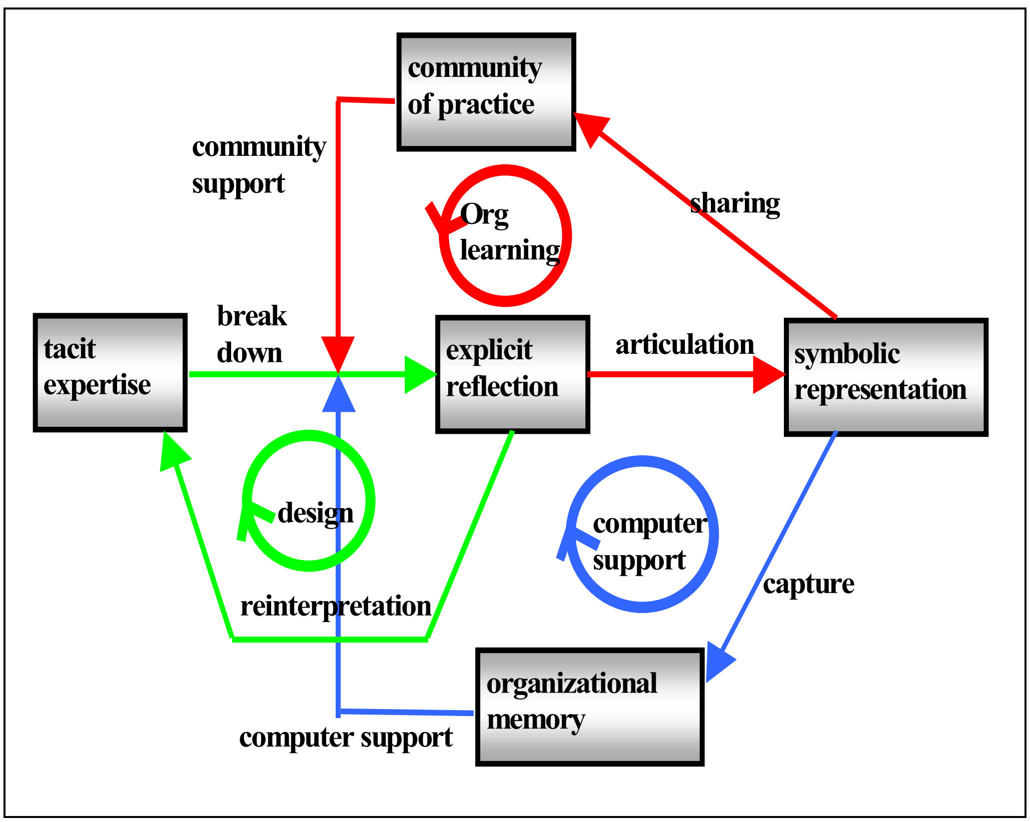


Figure 2. Cycles of design, computer support, and organizational learning.

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 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 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 *organizational learning* (see Figure 2, upper cycle). The personal reflection and collaborative articulation of shared perspectives makes innovation possible (Boland & Tenkasi, 1995; Tomasello et al., 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 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 organizational learning in communities of practice.

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., 1993/1998). 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 3, left).

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 a rich information space there may be many relationships of interest between new work artifacts and items in the 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 from the *task representation* to relevant *community memory* information items to support *learning on demand* (see Figure 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 & 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.

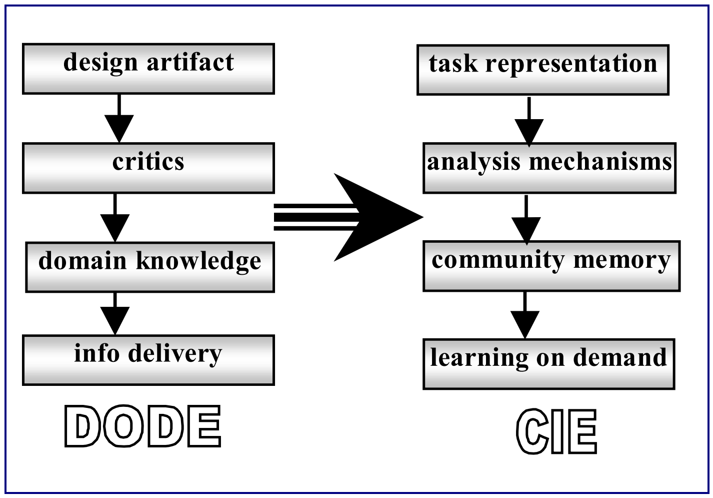


Figure 3. Generalization of the DODE architecture (left) to a CIE (right).

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.

WebNet: Scenario of a CIE for Design

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 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.

When Kay runs the LAN simulation, WEBNET proactively inserts a router (see Figure 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 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.

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.

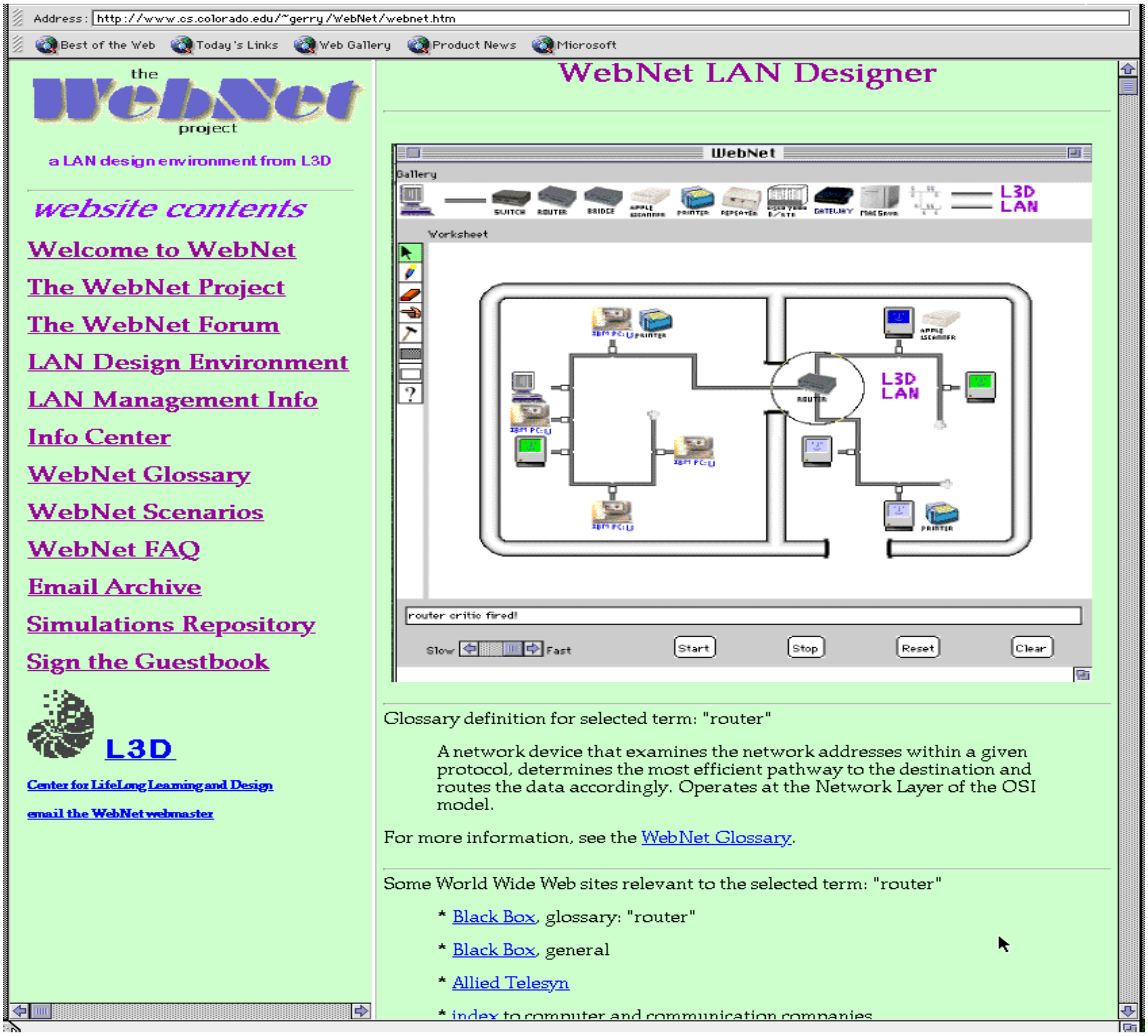


Figure 4. The *WEBNET* LAN design and simulation workspace (upper-right frame) and information delivered by a critic (lower-right frame). Note table of contents to the Web site (left frame).

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.

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.

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.

SECTION 3. 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. 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).

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 & 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 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 & 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 matches 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, 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 (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 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. We 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 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 workspaces (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 & 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.

Designing a System for Collaborative Knowledge Construction

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

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 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 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 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 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, organizing, and relating ideas as part of the collaborative construction of knowledge.

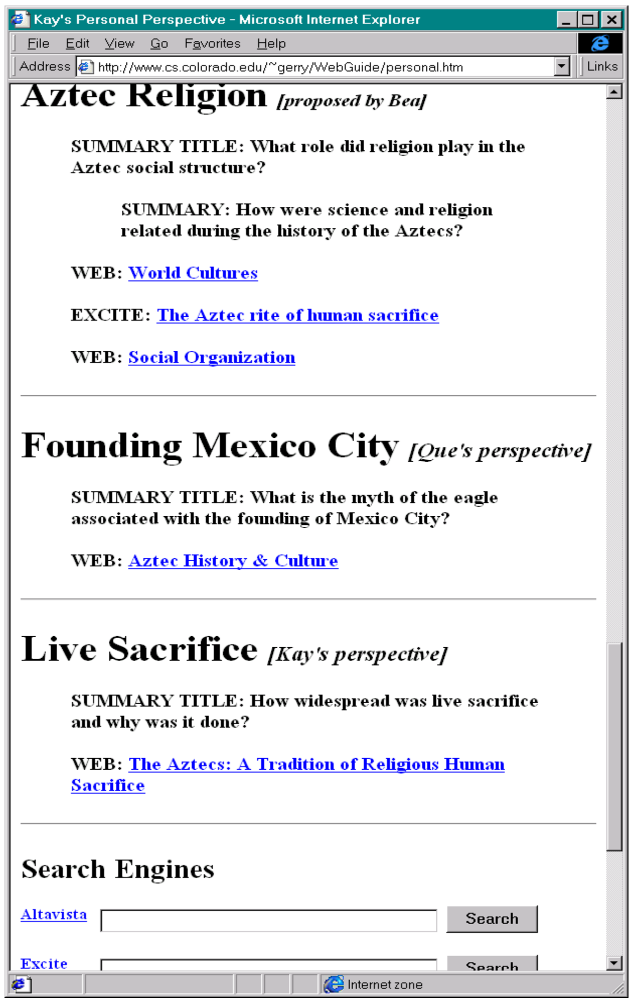


Figure 5. Part of Kay's personal perspective. There are three topics visible in this view. Within each topic are short subheadings or comments, as well as Web bookmarks and search queries. At the bottom is access to search engines.

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.

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 Figure 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 & 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 labeled by their perspective of origin. Students extract from the 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 group 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.

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 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 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.

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 Figure 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

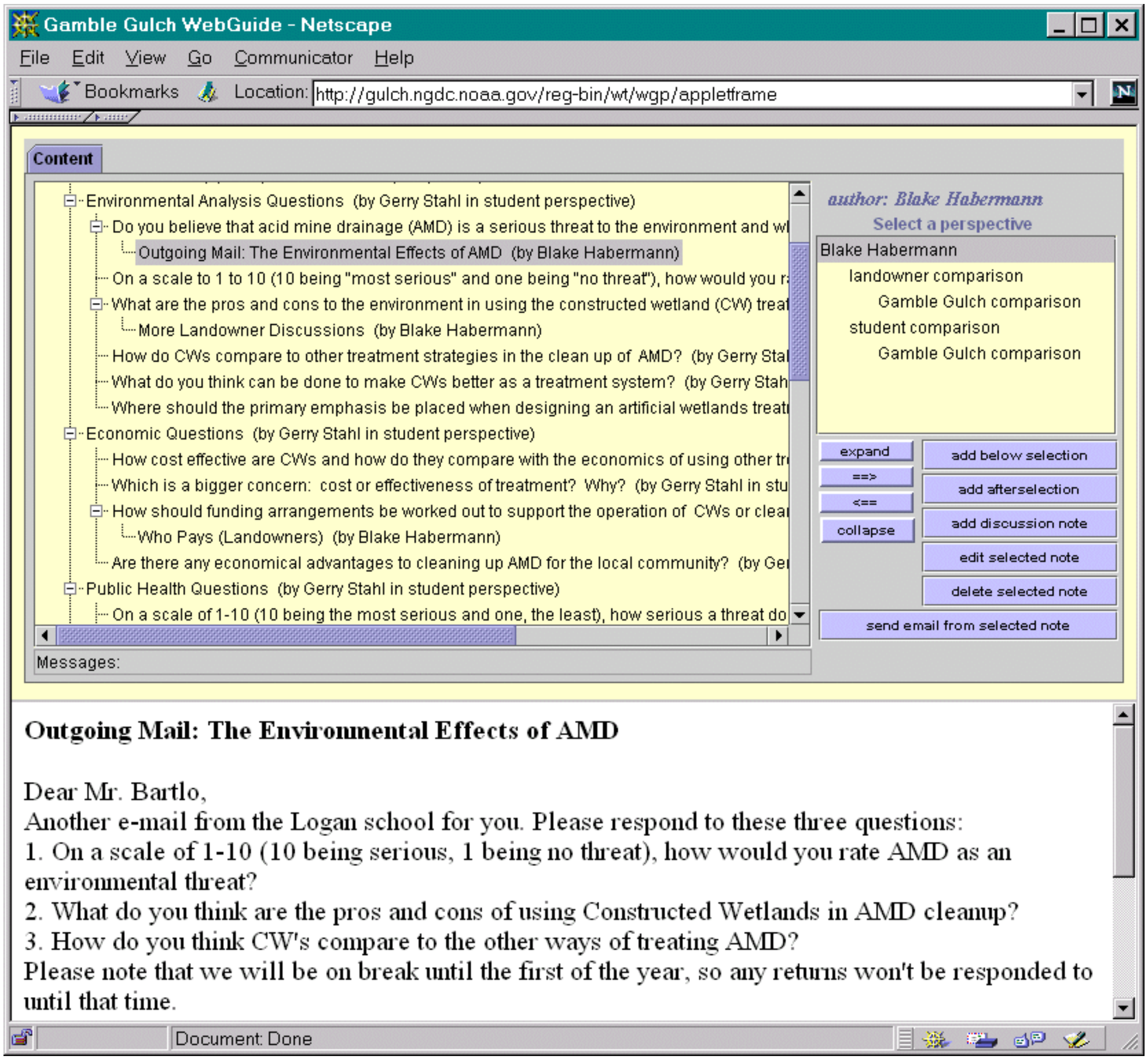


Figure 6. *WebGuide* for negotiating environmental perspectives.

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 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 Figure 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.

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 analyze*. 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 add, edit, or delete notes there (as in Figure 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 sub-groups 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 organization 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).

EXTENDING HUMAN COGNITION

Our early work on domain-oriented design environments (DODEs) – reviewed in Section 1 – was an effort to augment human intelligence within the context of professional design activities. At a practical level, our focus on

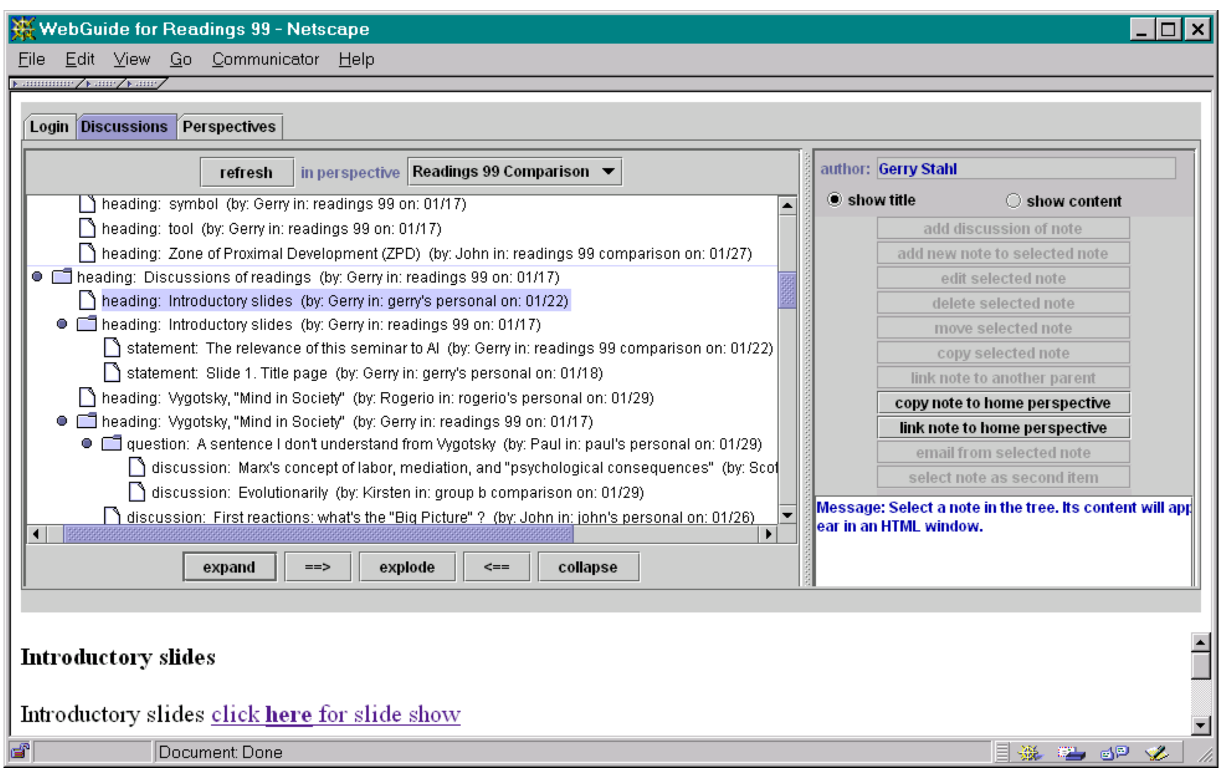


Figure 7. *WEBGUIDE* for constructing knowledge based on readings.

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, 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 2 – 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 & Duguid, 1991; Lave & 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 artifactual and the social (Orlikowski et al., 1995; Vygotsky, 1930/1978). CIEs are designed to serve as socially-embued, 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 3 – 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, reorganize 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 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). We are trying to explore the positive implications of this view by designing collaborative information environments to support knowledge construction by communities.

ACKNOWLEDGMENTS

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. & McDonald, D. W. (1996) Answer Garden 2: Merging organizational memory with collaborative help, *CSCW '96*, pp. 97-105.

Argyris, C. & 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., & Palincsar, A. (1991) Motivating project-based learning: Sustaining the doing, supporting the learning., *Educational Psychologist,* 26, pp. 369-398.

Boland, R. J. & Tenkasi, R. V. (1995) Perspective making and perspective taking in communities of knowing, *Organization Science,* 6(4), pp. 350-372.

Borghoff, U. & Parechi, R. (1998) *Information Technology for Knowledge Management,* Springer Verlag, Berlin, Germany.

Bourdieu, P. (1972) *Outline of a Theory of Practice,* Cambridge University Press, Cambridge, UK.

Brown, J. S. & Duguid, P. (1991) Organizational learning and communities-of-practice: Toward a unified view of working, learning, and innovation, *Organization Science,* 2(1), pp. 40-57.

Buckingham Shum, S. (1998) Negotiating the construction of organizational memories. In U. Borghoff & R. Parechi (Eds.), *Information Technology for Knowledge Management,* Springer Verlag, Berlin, Germany, pp. 55-78.

Buckingham Shum, S. & Hammond, N. (1994) Argumentation-based design rationale: What use at what cost?, *International Journal of Human-Computer Studies,* 40(4), pp. 603-652.

Bush, V. (1945) As we may think, *Atlantic Monthly,* 176(1), pp. 101-108.

Crook, C. (1994) Computers and the Collaborative Experience of Learning, Routledge, London, UK.

Donald, M. (1991) Origins of the Modern Mind: Three Stages in the Evolution of Culture and Cognition, Harvard University Press, Cambridge, MA.

Fischer, G., Grudin, J., McCall, R., Ostwald, J., Redmiles, D., Reeves, B., & Shipman, F. (1997) Seeding, evolutionary growth and reseeding: The incremental development of collaborative design environments. In G. M. Olson, T. W. Malone, & J. B. Smith (Eds.), *Coordination Theory and Collaboration Technology,* Lawrence Erlbaum Associates, Mahwah, NJ, p. in press.

Fischer, G., Lemke, A. C., McCall, R., & Morch, A. (1996) Making argumentation serve design. In T. Moran & J. Carrol (Eds.), *Design Rationale: Concepts, Techniques, and Use,* Lawrence Erlbaum and Associates, Mahwah, NJ, pp. 267-293.

Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., & Sumner, T. (1993) Embedding computer-based critics in the contexts of design, *Conference on Human Factors in Computing Systems (INTERChi '93)*, pp. 157-164.

Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., & Sumner, T. (1993/1998) Embedding critics in design environments. In M. Maybury & W. Wahlster (Eds.), *Readings in Intelligent User Interfaces,* Morgan-Kaufmann, SanFrancisco, CA, pp. 537-561.

Floyd, C. (1992) Software development and reality construction. In C. Floyd, H. Zuellinghoven, R. Budde, & R. Keil-Slawik (Eds.), *Software Development and Reality Construction,* Springer Verlag, Berlin, Germany, pp. 86-100.

Fu, M. C., Hayes, C. C., & 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), pp. 60-68.

Gadamer, H.-G. (1960/1988) *Truth and Method,* Crossroads, New York, NY.

Giddens, A. (1984) *The Constitution of Society,* U of California Press, Berkeley, CA.

Grudin, J. (1990) Why CSCW applications fail: Problems in design and evaluation, *CSCW '90*, pp. 85-93.

Habermas, J. (1981) *Theorie des kommunikativen Handelns, Volume 1,* Suhrkamp Verlag, Berlin, Germany.

Heidegger, M. (1927/1996) *Being and Time: A Translation of Sein und Zeit,* (J. Stambaugh, Trans.), SUNY Press, Albany, NY.

Koschmann, T. (Ed.) (1996) *CSCL: Theory and Practice of an Emerging Paradigm,* Lawrence Erlbaum Associates, Hillsdale, NJ.

Landauer, T. K. & 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), pp. 211-240.

Lave, J. & 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*, pp. 14-18.

Lindstaedt, S. & Schneider, K. (1997) Bridging the gap between face-to-face communication and long-term collaboration, *International ACM SigGroup Conference on Supporting Group Work*, pp. 331-340.

Marx, K. (1867/1976) *Capital, Volume I,* (Ben Fowkes, Trans.), Penguin, New York, NY.

McCall, R., Bennett, P., d'Oronzio, P., Ostwald, J., Shipman, F., & Wallace, N. (1990) PHIDIAS: Integrating CAD graphics into dynamic hypertext, *European Conference on Hypertext (ECHT '90)*, pp. 152-165.

Miller, P. L. (1986) Expert Critiquing Systems: Practice-based Medical Consultation by Computer, Springer Verlag, Berlin, Germany.

Mittal, S., Bobrow, D., & Kahn, K. (1986) Virtual copies at the boundary between classes and instances, *Object-Oriented Programming Systems, Languages and Applications (OOPSLA '86)*, pp. 159-166.

Nelson, T. (1981) *Literary Machines,* Mindful Press, New York, NY.

Norman, D. A. (1993) *Things That Make Us Smart,* Addison-Wesley Publishing Company, Reading, MA.

O'Malley, C. (1995) *Computer Supported Collaborative Learning,* Springer Verlag, Berlin, Germany.

Orlikowski, W. (1992) Learning from Notes: Organizational issues in groupware implementation, *CSCW '92*, pp. 362-369.

Orlikowski, W., Yates, Y., Okamura, K., & Fujimoto, M. (1995) Shaping electronic communication: The metastructuring of technology in the context of use, *Organization Science,* 6(4), pp. 423-369.

Orr, J. (1990) Sharing knowledge, celebrating identity: War stories and community memory in a service culture. In D. S. Middleton & D. Edwards (Eds.), *Collective Remembering: Memory in Society,* SAGE Publications, Beverly Hills, CA.

Polanyi, M. (1962) *Personal Knowledge,* Routledge & Kegan Paul, London, UK.

Repenning, A. (1994) Programming substrates to create interactive learning environments, *Journal of Interactive Learning Environments, Special Issue on End-User Environments,* 4(1), pp. 45-74.

Rittel, H. & Webber, M. M. (1984) Planning problems are wicked problems. In N. Cross (Ed.) *Developments in Design Methodology,* John Wiley & Sons, New York, NY, pp. 135-144.

Robbins, J. E. & Redmiles, D. F. (1998) Software architecture critics in the Argo design environment, *Knowledge-Based Systems*, p. in press.

Scardamalia, M. & 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, pp. 37-68.

Scardamalia, M. & Bereiter, C. (1996) Computer support for knowledge-building communities. In T. Koschmann (Ed.) *CSCL: Computer Supported Collaborative Learning,* Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 249-268.

Schön, D. A. (1983) The Reflective Practitioner: How Professionals Think in Action, Basic Books, New York, NY.

Senge, P. (1990) The Fifth Discipline, The Art and Practice of the Learning Organization, Currency Doubleday, New York, NY.

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, *Proceedings of the Cognitive Science Society (CogSci '93)*, pp. 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, *AERA '99*.

Stahl, G., Sumner, T., & 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), pp. 237-246.

Tomasello, M., Kruger, A. C., & Ratner, H. (1993) Cultural learning, *Behavioral and Brain Sciences*, pp. 495-552.

Vygotsky, L. (1930/1978) *Mind in Society,* Harvard University Press, Cambridge, MA.

Winograd, T. & Flores, F. (1986) Understanding Computers and Cognition: A New Foundation of Design, Addison-Wesley, Reading, MA.

Yates, J. & Orlikowski, W. (1992) Genres of organizational communication: A structurational approach to studying communication and media, *Academy of Management Review,* 17(2), pp. 299-326.

Zuboff, S. (1988) *In the Age of the Smart Machine,* Basic Books, Inc., New York, NY.

WEBGUIDE: GUIDING COLLABORATIVE LEARNING ON THE WEB WITH PERSPECTIVES

ABSTRACT

We are developing a Web-based tool called WEBGUIDE to mediate and structure collaborative learning. This software uses an innovative mechanism to define a flexible system of *perspectives* on a shared knowledge construction space. WEBGUIDE provides an electronic and persistent workspace for individuals and teams to develop and share distinctive points of view on a topic. We are designing the software and associated usage practices by trying it out in a middle school classroom and an advanced graduate seminar. Our experience in these use situations has raised a range of questions concerning theoretical and practical issues, which are driving our research. This paper is a reflection on what we are learning collaboratively about how software artifacts can mediate learning and shared cognition.

INTRODUCTORY NARRATIVE

For some years now I have been interested in how to personalize the delivery of information from knowledge repositories to people based on their preferred *perspectives* on the information (Stahl, 1995; 1996). For instance, designers often critique an evolving design artifact from alternative technical points of view; different designers have different personal concerns and styles, requiring considerations based upon access to different rules of thumb, rationale, constraints, standards and other forms of domain knowledge. Computer design environments should support these important interpretive perspectives (Stahl, 1993a; 1993b). I am now primarily interested in applying similar mechanisms of perspectival computer support within contexts of collaborative learning (Stahl, 1999).

Last year, Ted Habermann – an information architect at NOAA who makes geophysical data available to school children over the Web – suggested to me that we try to develop some computer support for a project at his son's middle school. Dan Kowal, the environmental sciences teacher at the Logan School for Creative Learning in Denver, was planning a year-long investigation of alternative perspectives on the issue of "acid mine drainage" (AMD) – the pollution of drinking water supplies by heavy metals washed out of old gold mines. The fact that Dan and I were interested in "perspectives" from different perspectives seemed to provide a basis for fruitful collaboration. Ted obtained NSF funding for the project and we all spent last summer planning the course and its perspectives-based software. Each of us brought in colleagues and worked to create a Java application (WEBGUIDE), a set of auxiliary web pages, a group of adult mentors representing different perspectives on AMD and a course curriculum.

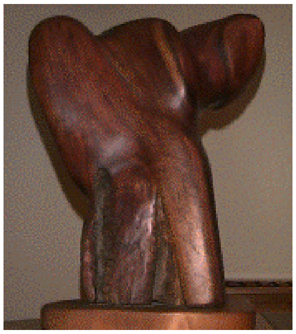
The class started in September and the software was deployed in October. The students in Dan's class were aware of the experimental nature of the software they were using and were encouraged to critique it and enter their ideas into WEBGUIDE. Feedback from these twelve-year-old students provided initial experience with the usability of WEBGUIDE and resulted in a re-implementation of the interface and optimization of the algorithms over Christmas vacation.

In January, I organized an interdisciplinary seminar of doctoral students from cognitive, educational and computational sciences to study theoretical texts that might provide insight into how to support collaborative learning with perspectives-based software. The seminar uses WEBGUIDE as a major medium for communication and reflection, including reflection on our use of the software. This provides a second source of experience and raises a number of issues that will need to be addressed in software redesign this summer.

In this paper I would like to begin a reflection on the issues that have arisen through our WEBGUIDE experiences because I think they are critical to the ability to support collaborative learning with computer-based environments. The potential for computer mediation of collaboration seems extraordinary, but our experience warns us that the practical barriers are also enormous. Certainly, our experiences are not unique, and similar projects at the universities of Toronto, Michigan, Berkeley, Northwestern, Vanderbilt, Georgia Tech, etc. have run into them for years. Indeed, we observed many of these issues in a seminar last year prior to the implementation of WEBGUIDE (dePaula, 1998; Koschmann & Stahl, 1998). However, I believe that perspectives-based software addresses or transforms some of the issues and raises some of its own.

Now let me describe our experience in the two situations of classroom practice and explain the underlying computational perspectives approach.

Practice I: Environmental Perspectives



An early implementation of WebGuide is in use in Dan's classroom at the Logan School. For the past five years, his class of middle school students has researched the environmental damage done to mountain streams by "acid mine drainage" from deserted gold mines high in the Rocky Mountains above Denver. The students actually solved the technical problem at the source of a stream coming into Boulder from the Gamble Gulch mine site by building an artificial constructed wetlands area to filter out heavy metals. This year they are investigating the broader ramifications of their success; they are looking at the social 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 landowner. Then, working in teams corresponding to each of these perspectives, they articulate the position of their perspective on a set of shared questions.

The "Gamble Gulch" application of WebGuide serves as the medium through which the students collaboratively research these issues with their mentors and with teammates. Each student and mentor has their personal display perspective, and their display perspectives each inherit from one of the content-based team perspectives (environmental protection, governmental regulation, etc.), depending upon which intellectual perspective they are working on constructing.

Figure 1 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 AMD is a serious threat to the environment?" Here, Blake has sent an email to a mentor asking for information related to this question. Email interactions happen through WebGuide and are retained as notes in its display perspectives. When replies are sent back, they are automatically posted to the discussion outline under the original email. When someone clicks on a title, the contents of that note are displayed in an HTML frame below the applet (as is the body of the student's email in Figure 1).

Blake is working in his personal perspective, which inherits from the class, student team and landowner team perspectives (see the red arrows in Figure 2). Note that the display of his personal perspective (in Figure 1) includes notes that Dan and I entered in the student perspective to structure the work of all the students. Blake 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 comparison, the Landowner comparison and the Gamble Gulch class comparison perspectives (see list of perspectives accessible to him on the right of Figure 1).

For this application, the teacher has decided that perspective comparing and negotiation 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. Dan 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.

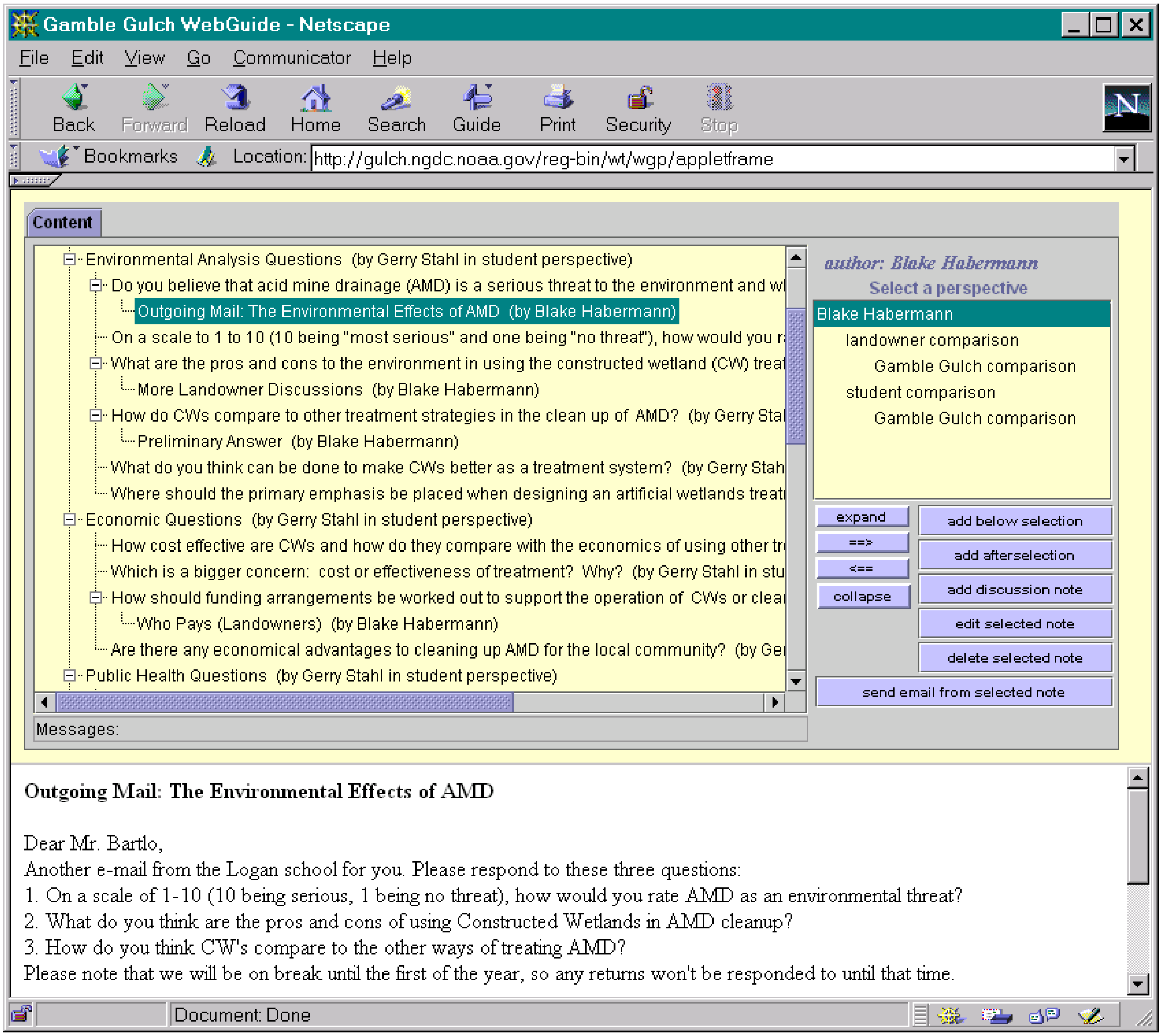


Figure 1. The Gamble Gulch version of *WebGuide* viewed in a Web browser. The top part is a Java applet displaying an outline view of note titles. The content of the selected note is displayed in an HTML frame below. To the right are buttons for navigating the outline and changing the content in the shared knowledge space. The view shown is from the personal perspective of one student.

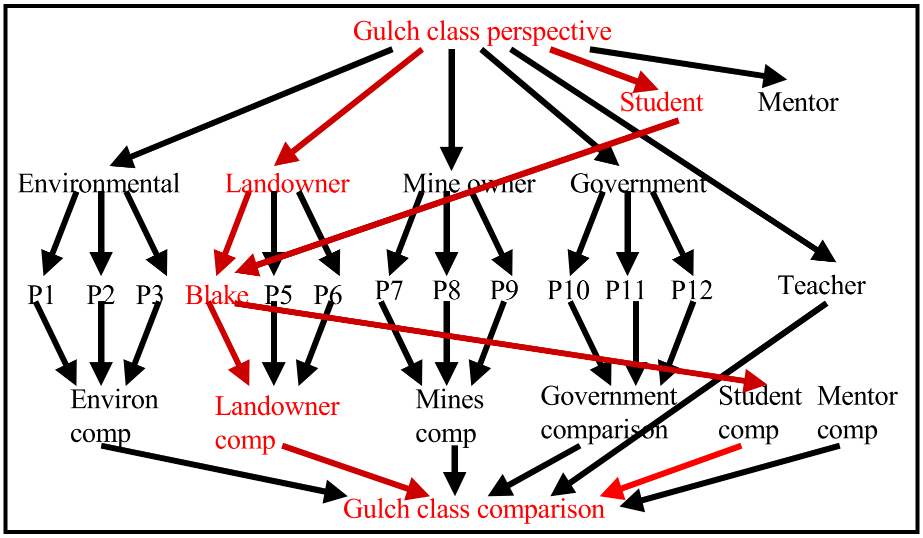


Figure 2. The web of perspectives in Gamble Gulch. Information is automatically inherited downward in the diagram. Blake's perspective includes all the notes entered in the Gulch class, Landowner and Student perspectives. His notes also show up in the Landowner, Student and Gulch class comparison perspectives.

DEFINITION OF PERSPECTIVES

The term "perspectives" is over-loaded with meanings; this frequently produces confusion even when it is intended to tacitly exploit aspects of the perspectives metaphor from one domain into another. It may be helpful at this point to distinguish three types of perspectives: literal, figurative and computational.

- *Literal perspectives* are optical or perceptual orientations: one sees objects from the specific angle or vantage point of the physical location of ones eyes.
- *Figurative perspectives* take metaphorical license and refer to, for instance, different ways of conceptualizing a theme, as in adopting a skeptical view of a conversational claim.
- *Computational perspectives* are the result of software mechanisms that classify elements in a database for selective display. In WEBGUIDE, for example, if I enter a note in my personal perspective then that note will be displayed whenever my perspective is displayed but not when someone else's personal perspective is displayed.

WEBGUIDE implements a system of computational perspectives designed to exploit the perspective metaphor in order to support characteristics of collaboration and collaborative learning. It is unique in a number of ways that distinguish it from other software systems that may use the term "perspectives":

Other systems refer to different *representations* of information as perspectives. They might have a graphical and a textual view of the *same* data. In WEBGUIDE, different data is displayed in different perspectives – using the same representation, hierarchically structured titles of textual notes.

- In WEBGUIDE, the perspectives mechanism is neither a simple tagging of data nor a database view, but is a dynamic computation that takes into account a *web of inheritance* among perspectives. Thus, Blake's perspective includes not only information that he entered in his perspective, but also information inherited from the Class, Student and Landowner perspectives.

- Furthermore, the web of perspectives can be extended by users *interactively* and the inheritance of information is always computed based on the current configuration of this web.
- In addition, the information in a perspective has a user-maintained structure in which each note has one or more parent notes and may have children notes, creating a *web of notes* within each perspective. The order of children displayed under a parent note is user-defined and maintained so that WEBGUIDE can be used to *organize ideas* within outline structures.

The idea of perspectives on the Web traces its lineage to 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 hypertext spaces. At the University of Colorado we have been exploring this approach to computational perspectives in desktop applications for the past decade (McCall et al., 1990; Stahl, 1993b). WEBGUIDE is our first truly Web-based version. The core of WEBGUIDE consists of a perspectives server named POW! (Perspectives On the Web), which communicates with Java, Perl or HTML interfaces.

The computational perspectives mechanism we have been exploring incorporates the following features for a community of users (Stahl, 1993a):

- Individual community members have access to what appears to be their own information source. This is called their *personal perspective*. It consists of notes from a shared central information repository that are tagged for display within that particular perspective (or in any perspective inherited by that perspective).
- Notes can be created, edited, rearranged, linked together or deleted by users within their own personal perspective without affecting the work of others.
- Another student, Annie, can integrate a note from Blake's perspective into her own personal perspective by creating a *link* or virtual copy of the note. If Blake modifies the original note, then it changes in Annie's perspective as well. However, if Annie modifies the note, a new note is actually created for her, so that Blake's perspective is not changed. This arrangement generally makes sense because Annie wants to view (or inherit) Blake's note, even if it evolves. However, Blake should not be affected by the actions of someone who copied one of his notes.
- Alternatively, Annie can *physically copy* the contents of a note from Blake's perspective. In this case, the copies are not linked to each other in any way. Since Annie and Blake are viewing physically distinct notes now, either can make changes without affecting the other's perspective.
- There is an inheritance web 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. 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. A hierarchy of team, sub-team and individual perspectives can be built to match the needs of a particular community.

This model of computational 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 figurative perspective. Such "perspective-making" is important in thinking about and judging issues from particular perspectives.

However, if one wants to influence the content of other team members' perspectives through "perspective-taking" (Boland & Tenkasi, 1995), then this approach is limited because one cannot change someone else's content directly. Moreover, for supporting collaborative work it is important 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 (Tomasello et al., 1993). We are interested in exploring how to support the intertwining of perspectives with our computational perspectives mechanisms. We will return to this issue after describing the types of perspectives used in our applications.

TYPES OF PERSPECTIVES

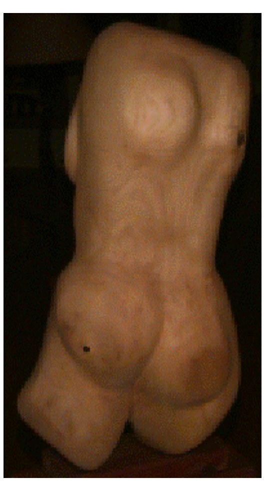
WEBGUIDE provides several levels of perspectives within a web of perspective inheritance to help students compile their individual and joint research:

- The *class perspective* is created by the teacher to start each team off with an initial structure and some suggested topics. It typically establishes a framework for classroom activities and defines a space used to instantiate the goal of collecting the products of collaborative intellectual work.
- The *team perspective* contains notes that have been accepted by a team. This perspective can be 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 student's team's perspective. Thus, it displays the owner's own work within the context of notes 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 notes in their personal perspectives. They can also create completely new material there. This computational perspective provides a personal workspace in which a student can construct his or her own figurative perspective on shared knowledge.
- 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 in the team. It inherits from the personal, team and class perspectives. Students can go here to get ideas and copy notes into their own personal perspective or propose items for the team perspective.

Of course, there is not really a duplication of information in the community memory. The perspectives mechanism merely displays the information differently in the different perspectival views, in accordance with the relations of inheritance.

To design software for collaborative learning in schools means to design curriculum and classroom process as well (Stahl et al., 1995a; 1995b). Computer support has to be matched with appropriate content (typically stored in WEBGUIDE or on the Web) and with constructivist practices for knowledge-building communities (Scardamalia & Bereiter, 1991). The design of the WEBGUIDE interface and the perspectives mechanism must be adapted to individual application situations, with appropriate seeding of content, structuring of the perspectives web and establishing of access policies.

In Logan School, for instance, students each enter notes in their personal perspectives using information available to them: from the Web, books, encyclopedia, discussions, interviews of mentors or other sources. Students can review the notes in the class perspective, their team perspective and the personal perspectives of their teammates. All of these contents are collected in comparison perspectives, where they are labeled by their perspective of origin. Students extract from the research those items that are of interest to them. Then they organize and develop the data they have collected by categorizing, summarizing, labeling and annotating. The stages of investigating, collecting and editing can be iterated as many times as desired. Team members then negotiate which notes should be promoted to the team perspective to represent their collaborative statement of their perspective on acid mine drainage.



ISSUES FOR PERSPECTIVES

As an initial field testing of the WEBGUIDE system, the Logan School trial is generating 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 (within 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.

The first issues to hit home when we deployed WEBGUIDE were the problems of response time and screen real estate. The student computers were slower, had smaller monitors, lacked good Internet connections and were further from the server than the computers of the developers. We were, of course, already familiar with these issues from other Web applications, but one never knows quite how things will work out and how they will be accepted until one tests them under classroom conditions.

A pre-release prototype of WEBGUIDE used dynamic HTML pages. This meant that each time one expanded a different part of the outline of titles one had to wait for a new page to be sent across the Internet. It also greatly constrained the interface functionality. However, when we moved to a Java applet, we had to wait several minutes to download the applet code to each student computer. Furthermore, it entailed running all the perspectives computations on the slow student computer. In order to reduce the download time significantly, we first rewrote the interface using standard Java Swing classes that can be stored on the student machines. Then we split the applet into a client (the interface) and a server (the perspectives computations and database access). By downloading only the client part to the classroom, we not only reduced the download time further, but also ran the time-consuming computations on our faster server computers.

Such technical problems can be solved relatively easily, by optimizing algorithms or by adjusting tradeoffs based on local conditions. Issues of social practice are much more intransigent. There seem to be two major issues for software like WEBGUIDE, that is, software for threaded discussions and collaborative knowledge construction:

- Lack of convergence among the ideas developed in the supported discussions.
- Avoidance of system use in favor of email, face-to-face conversation or inaction.

WEBGUIDE introduces its computational perspectives mechanism as a structural feature to facilitate the articulation of convergent ideas and even incorporates email. In attempting to address the above problems, it raises a new set of issues:

- Is the perspectives metaphor a natural one (or can it be made natural) so that people will use computational perspectives to construct their figurative perspectives?
- Can the web of perspectives be represented in a convenient and understandable format?

In our trials of WEBGUIDE we have tried to create learning situations that would encourage the use of the software, yet we have observed low levels of usage and under-utilization of the system's full functionality. This raises the following additional issues:

- How can learning situations be structured to take better advantage of the presumed advantages of the software?
- How can the system's various capabilities be distinguished, such as its support for threaded discussions and for perspective-making?

In order to answer questions of this magnitude it was necessary to gather more experience, to be more closely involved in the daily usage of the system and to develop a deeper theoretical understanding of collaborative learning and of computer mediation. Having defined these goals, I announced a seminar on the topic of "computer mediation of collaborative learning," open to interested researchers from a number of disciplines – primarily education, cognitive psychology and computer science. The goal of the seminar was explicitly stated to be an experiment in the use of WEBGUIDE to construct knowledge collaboratively, based on careful reading of selected texts. The texts traced the notion of *computer mediation* (Boland & Tenkasi, 1995; Caron, 1998; Hewitt et al., 1998; Scardamalia & Bereiter, 1996; Stahl, 1999) back to *situated learning theory* (Bruner, 1990; Cole, 1996; Lave, 1991; 1996; Lave & Wenger, 1991) – and from there back to the notion of *mediated consciousness* in Vygotsky (1930/1978) and its roots in Hegel (Habermas, 1971; Hegel, 1807/1967; Koyeve, 1947/1969) and Marx (1844/1967; 1845/1967; 1867/1976).

In the final section of this paper I will comment on our current understanding of the six issues listed above. But first it is necessary to describe the ways in which the seminar attempts to make use of WEBGUIDE and the conceptualization of the theory of computer mediation that is arising in the seminar.

PRACTICE II: THEORETICAL PERSPECTIVES

The seminar on computer mediation of collaborative learning is designed to use WEBGUIDE in several ways:

- *As the primary communication medium for 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 collaboration support system to analyze*. 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.
- *As a glossary and reference library*. This application of WEBGUIDE is seeded with a list of terms that are likely to prove important to the seminar and with the titles of seminar readings. Seminar members can develop their own definitions of these terms, modifying them based on successive readings in which the terms recur in different contexts and based on definitions offered by other members. Similarly, the different readings are discussed extensively within WEBGUIDE. This includes people giving their summaries of important points and asking for help interpreting obscure passages. People can comment on each other's entries and also revise their own. Of course, new terms and references can be added easily by anyone.
- *As a brainstorming arena for papers*. The application has already been seeded with themes that might make interesting research papers drawing on seminar readings and goals. WEBGUIDE allows people to link notes from anywhere in the information environment to these themes and to organize notes under the themes. Thus, both individuals and groups can use this to compile, structure and refine ideas that may grow into publishable papers. Collaborative writing is a notoriously difficult process which generally ends up being dominated by one participant's perspective or being divided up into loosely connected sections, each representing a single perspective. WEBGUIDE may facilitate a more truly collaborative approach to organizing ideas on a coherent theme.
- *As a bug report mechanism or feature request facility*. Seminar participants can communicate problems they find in the software as well as propose ideas they have for new features. By having these reports and proposals shared within the WEBGUIDE medium, they are communicated to other seminar participants, who can then be aware of the bugs (and their fixes) and can join the discussion of suggestions.

The seminar version of WEBGUIDE incorporates a built-in permissions system that structures the social practices surrounding the use of the system. Seminar participants each have their own 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 personal 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 ones own perspective then one cannot add, edit or delete notes there (as in Figure 3). To manipulate notes freely, one must first copy or link the note into ones 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 sub-groups 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 class 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 organization of a single perspective. This supports the taking of other people's perspectives and the use of shared ideas in the making of ones own perspectives (Boland & Tenkasi, 1995).

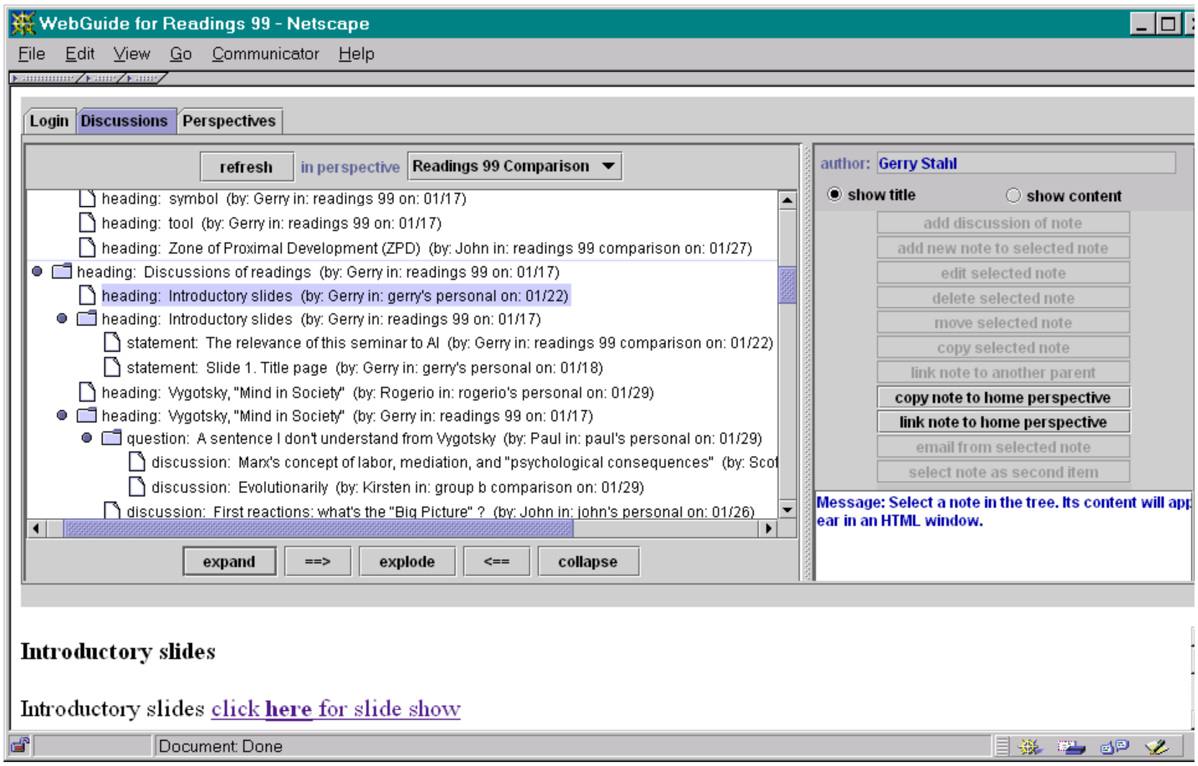
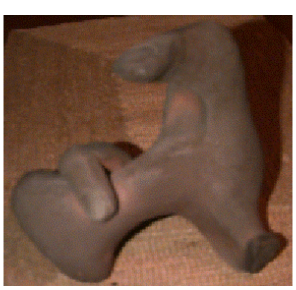


Figure 3. The version of *WebGuide* used in the seminar. Note that some of the control buttons on the right are not functional when the logged-in author is not working in his own personal perspective. This enforces certain social practices. Also note that many headings have been inserted to structure the discussion space.

The seminar application of WebGuide stresses the use of perspectives for structuring collaborative efforts to build shared knowledge. The goal of the seminar is to evolve theoretical views on computer mediation – and to do so within a medium that supports the sharing of tentative positions and documents the development of ideas and collaboration over time. A major hypothesis investigated by the seminar is that software environments with perspectives – like WebGuide – can provide powerful tools for coordinated intellectual work and collaborative learning. It explores how the use of a shared persistent knowledge construction space can support more complex discussions than ephemeral face-to-face conversation. Many of the desires and concerns in this paper arose in notes in WebGuide as part of the seminar. In particular, the seminar's focus on theory as our practice has problematized our understanding of the role of theory.

Theory in Practice



Our initial application of WebGuide in the middle school environmental course raised a number of issues that led us to seek theoretical understanding through a seminar, which is serving as a second application of WebGuide. We have begun to see our research differently as a result of the theories we are incorporating in our reflections within the seminar. One thing that has changed is the relation we see of this theory to our research practice.

In my paper proposal to AERA, written prior to our recent explorations, I described our approach by following the narrative order implied by conventional wisdom about the relation of theory to practice. After stating the goal or purpose of the work, I provided a theoretical framework, followed by sections on techniques, evidence, conclusions and educational / scientific import. The assumption here was that when one had a problem one turned first to theory for the solution and then "applied" the theory to some situation – either the problem situation or an experimental test context. After designing the solution based on the pre-existing theory and applying it to the test situation, one gathered evaluative data and analyzed the data to measure success. The evaluation then implies whether or not the solution has generalizable import.

But such an approach is in keeping neither with our current experience nor with our emerging theory. We started last summer with an opportunity to explore some vague notions we had about something we called "perspectives". We experimented with ever-evolving techniques through a complex collaborative process involving many people, each with their own concerns, understanding and insights. As part of this process some of us turned to theory – but the selection of theoretical texts and our interpretations of them were determined by the processes and issues we observed in our practical strivings.

So in this draft of the paper – still not considered a static final document, but a recapitulation from one particular moment in an on-going process – I am trying to narrate a different story about how theory and practice have been co-mingled in our research. We began with an idea for a concrete classroom curriculum and worked on designing tools and structures to support the practical needs of that curriculum. Once we had a working software prototype that could be used over the Web, we deployed it in the middle school classroom. We immediately confronted the realities of issues of response speed and monitor screen real estate that we had been worried about from the start. Students started asking for new functionality and it became clear that they were not using the implemented functions the way they were designed to be used. A dance commenced between the technicians, the educators, the students, the curriculum and the software; as we circled each other, we changed and became more compatible with each other.

There was no point in trying to evaluate the success of our experiment by gathering data under controlled conditions. It was clear that we needed to figure out how to make things work better, not to measure precisely how well they were (or were not) already working. Beyond the relatively clear technical usability issues there were deeper questions of how software can mediate interpersonal and cognitive relations within collaboration (Hewitt et al., 1998). This led us to look for a theory of computer mediation – and for that matter a theory of collaborative learning – in the graduate seminar. Of course, it turned out that there are no adequate theories on these topics sitting on the bookshelf for us to simply apply. Rather, we had to undertake the construction of such theory, building upon hints strewn around in texts from many disciplines and guided by the problematic in which we are involved first hand.

Trusting in our intuition that software like WEBGUIDE could facilitate group theory building, we set out to use WEBGUIDE in our theoretical investigations, and thereby drive the further development of the software through additional practical experience even as we were developing theoretical justifications for our design. In reflecting on our experience, I have tried to organize this draft of the paper in accordance with a non-traditional theory about the relation of theory and practice – an understanding of this relationship more in keeping not only with our practice but with our hermeneutic, dialectical, socially situated activity theory.

Thus, we started out from our vague, only partially articulated background understanding of perspectives as an interesting and promising concept for learning and for computer support (Stahl, 1999). We set up a real-world situation in which we could explore what happens. In this situation we nurtured a process of "structural coupling" (Maturana & Varela, 1987) in which the different actors evolve toward a workable synthesis or homeostasis. Rapid prototyping cycles and participatory design sessions help facilitate this process. As breakdowns in how things were intended to work are recognized, we engage in reflection-in-action (Schön, 1983) to make our tacit pre-understanding explicit, to understand what has happened and to project corrective actions. This process of explication raises generalizable issues and calls for theory. But despite the generality of the issues, the theory is not understood in a completely abstract way, but in terms of its relevance to our situation and to the specific barriers we have uncovered in that concrete situation (Stahl, 1993a).

Theory – like everyday thought – often arises after the fact (or well into the complex process of practical investigations) in order to justify situations that would otherwise be too messy to comprehend and remember. Then, first chance it gets, theory reverses the order of things and presents itself as a guiding *a priori*. As Hegel (1807/1967) says, "the owl of Minerva flies only at night": the wisdom of theory arrives on the scene only after the practical events of the day (which theory captures in concepts) have been put to bed. *Theory* is a cherished way to capture an understanding of what has been learned, even if it distorts the picture by claiming that the practice out of which theory arose was a simple application of the theory's pre-existing abstract principles.

But, as the analyses of mediated cognition our seminar is studying point out, there are other *artifacts* (Cole, 1996) in which experience can be captured, preserved and transmitted. *Narrative* is one (Bruner, 1990). In this paper I have tried to project a voice which does not redefine the temporality of the experience I am reporting. *Sculpture* is another way in which people impose meaningful form on nature and, as Hegel would say, externalize their consciousness through the mediation of wood, clay, plaster or stone – sharing it with others and preserving it as part of their culture's spirit. [1]

Polished *software* is a very different way of objectifying experience. Buried in the source code and affordances of a software artifact are countless lessons and insights – not only those of the particular software developer, but of the traditions (congealed labor) of our technological world upon which that developer built (Marx, 1867/1976). This is true of the current version of WEBGUIDE, as it is of any software application. But WEBGUIDE strives to preserve insights explicitly as well, within the notes displayed in its perspectives and within their organization, including their organization into personal and group perspectives. Perhaps when we understand better how to use WEBGUIDE in collaborative learning contexts it will maintain the knowledge that people construct through it in a way that preserves (*aufheben*) the construction process as well as the resultant theory. Eventually, collaborative practice and software design may co-evolve to the point where they can integrate the insights of multiple perspectives into group views that do not obliterate the insights of conflicting perspectives into the multifaceted nature of truth.

ISSUES FOR MEDIATION

We conclude this paper with an attempt to sort out what we are collaboratively learning through our use of WEBGUIDE. The six issues for perspectives-based software like WEBGUIDE that arose during the middle school application appeared in the graduate seminar's usage of the software as well – and were articulated by seminar participants in their notes in WEBGUIDE. These are important and complex issues that other researchers have raised as well. They are not problems that we have solved, but rather foci for future work. They define central goals for our redesign of WEBGUIDE this summer and goals for structuring the mediation of collaborative practices next year.

Here is a summary of our current understanding of these issues, based on our two practical experiences and our reflections on the theory of computer mediation of collaborative learning:

1. *Divergence among ideas*.

In his review of computer mediated collaborative learning, dePaula (1998) identified divergence of ideas to be a common problem. He argued that the tree structure imposed by standard threaded discussion support was inappropriate for collaboration. The idea of a threaded discussion is that one contribution or note leads to another, so that each new idea is connected to its "parent" in order to preserve this connection. The problem is that there is often no effective way to bring several ideas together in a summary or synthesis because that would require a particular note to be tied to several parent notes – something that is typically not supported by discussion software. The result is that discussions proceed along ever diverging lines as they branch out, and there is no systematic way to promote convergence. It seems clear, however, that collaboration requires both divergence (e.g., during brainstorming) and convergence (e.g., during negotiation and consensus).

WEBGUIDE tries to avoid this common structural problem of threaded discussion media at three levels: (1)The note linking mechanism in WEBGUIDE allows notes to be *linked* to multiple parents, so that they can act to bring together and summarize otherwise divergent ideas. As in threaded discussions, every note is situated in the workspace by

[1] See the images of sculptures throughout this paper. Of course, my sculptures are not the result of some primordial experience of self-consciousness interacting with unmediated nature. They are late twentieth century explorations of form and material. Here, organic three-dimensional forms are showcased to contrast with socially prevalent two-dimensional representations and with the geometric shapes produced by machinery. The characteristics of the materials of nature are brought forth, in contrast to the plastic substances that retreat from our consciousness in commodities. Also, the pragmatic representational function of symbolic objects is sublimated in the study of their abstracted physical forms and materiality. In negating the commonplace characteristics of signs – which point away from themselves – the non-representational sculptures obtrusively confront their creator and viewers with the nature of the artifact as intentionally formed material object.

being identified and displayed as the child of some other note. However, WEBGUIDE allows multiple parents, so that the web of notes is not restricted to a tree. (2) Similarly, the graph of perspectives allows for *multiple inheritance*, so that "comparison" perspectives can be defined that aggregate or converge the contents of multiple perspectives. The Logan School application was seeded with comparison perspectives corresponding to the class and subgroup perspectives, so that the overall perspectives graph has a structure in which the inheritance of notes first diverges from the class to the subgroup and then the personal perspectives, and then converges through the subgroup comparison perspectives to the class comparison perspective, as shown in Figure 2. The web of perspectives forms a directed acyclical graph rather than a strict hierarchy. (3) Another effective way to encourage a well-structured discussion is to seed the workspace with a set of headings to scaffold the discourse. By introducing carefully conceived *headings* high in the perspective inheritance network, a facilitator (such as a teacher) can define an arrangement of topics that will be shared by the participants and will encourage them to arrange related ideas close to each other.

Although WEBGUIDE provided these three convergence mechanisms in both of our usage situations, most participants were not adept at using any of them. This is probably related to the other issues below and is something that needs to be explored further in the future.

2. *Avoidance of system use*.

Media competition poses a barrier to acceptance of new communication software. People are naturally hesitant to adopt yet another communication technology. In a world inundated with pagers, cell phones, voicemail, email, fax, etc. people are forced to limit their media or be overwhelmed. They must calculate how much a burden the new medium will impose in terms of learning how to use it, acquiring the equipment, checking regularly for incoming messages and letting people know that they are communicating through it. Clearly, a *critical mass* of adoption by ones communication partners is necessary as well.

In a classroom context, some of these problems are minimized: all ones partners are required to use WEBGUIDE and the hardware is made available. Yet, it is not so simple. The Logan School students have to communicate with mentors who may not have Internet access or the proper hardware. Communication with classmates is much easier face-to-face then typing everything (knowing it has to be carefully done for grading). In the graduate seminar, most participants do not have convenient access to the necessary equipment and have to go out of their way to a special lab. This means that they are lucky to communicate through WEBGUIDE once a week, and therefore cannot enter into lively on-going interchanges.

This summer we will have to make WEBGUIDE more accessible by increasing the number of platforms/browsers that it can run on and making it work over slow modems from home. Further, we need to improve its look-and-feel to increase people's comfort level in wanting to use it: speed up response time, allow drag-and-drop rearrangement of notes, permit resizing of the applet and fonts for different monitors and different eyes, support searching and selective printouts, provide graphical maps of the webs of perspectives and nodes.

3. *Naturalness of the perspectives metaphor*.

Despite the fact that WEBGUIDE has been designed to make the perspectives metaphor seem natural and simple to navigate, people express confusion as to how to use the perspectives. What perspective should I be working in, browsing for other people's ideas or entering for discussions? The metaphor of perspectives as a set of alternative (yet linked and over-lapping) textual workspaces is a new notion when operationalized as in WEBGUIDE.

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 extremely 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. This is the goal of WEBGUIDE. Although the software now hides much of the complexity, it is not yet at the point where people can operate smoothly without worrying about the perspectives all together.

4. *Representation of the web of perspectives*.

One problem that aggravates acceptance of the perspectives metaphor is that the web of inheritance of content from perspective to perspective is hard to represent visually within WEBGUIDE. The WEBGUIDE interface relies on an *outline* display. This has many advantages, allowing users to navigate to and view notes of interest in an intuitive way that is already familiar. However, an outline display assumes a strictly hierarchical tree of information. Because the web of perspectives has multiple inheritance, its structure is not visible in an outline, which always shows a perspective under just one of its parents at a time. Thus, for instance, there is no visual representation of how a comparison perspective inherits from several personal perspectives.

The same is true at the level of notes. A note that has been linked to several other notes that it may summarize is always displayed as the child of just one of those notes at a time.

Two solutions suggest themselves for future exploration. One is to provide an alternative representation such as a *graphical map* in place of the outline view. As appealing as this sounds, it may be technically difficult to do on-the-fly. A bigger problem is that graphical maps are notoriously poor at scaling up. Already in our two trial situations – in which there are on the order of twice as many perspectives as participants – it would be hard to clearly label a graphical node for every perspective within the applet's confined display area. The second alternative is to indicate additional links with some kind of *icon* within the outline view. This would require more understanding on the part of the users in interpreting and making use of this additional symbolic information.

5. *Structuring of learning situations*.

We have argued based on previous experience that the crucial aspect of supporting collaborative learning has to do with structuring social practices (Koschmann et al., 1998). *Practice* in the sense of Bourdieu's concept of *habitus* (Bourdieu, 1972) is the set of generally tacit procedures that are culturally adopted by a community. In introducing WEBGUIDE into its two user communities, we have tried to establish certain usage practices, both by instruction and by enforcement in the software. Looking back at Figure 1, you can see that Logan students are only allowed to navigate to certain perspectives – namely their personal perspective and those group perspectives that inherit from that perspective. Seminar participants were originally given *permission* to navigate throughout the system and to make changes anywhere. That was subsequently modified (as shown in Figure 3) to restrict their abilities when not in their personal perspective. The governing principle was that everyone should be able to do anything they want within their personal perspective, but no one should be able to affect the display of information in someone else's personal perspective.

When the ability to enter notes everywhere was restricted, facilities for copying and linking notes from other computational perspectives into ones own computational perspective were introduced. This was intended to encourage people to integrate the ideas from other figurative perspectives into their own figurative perspective by making a conscious decision as to where the new note should go in their existing web of notes. However, this added a step to the process of communication. One could no longer simply select a note that one wanted to comment on and press the "add discussion" button.

In order to facilitate discussion of notes that one did not necessarily want to integrate into ones own perspective, the "add discussion" button was then made active in all comparison perspectives. This led to minor problems, in that one could then not edit discussion notes that one had contributed in these perspectives. This could be fixed at the cost of additional complexity in the rules by allowing the author of a note to edit it in comparison perspectives.

More significantly, our experiments with changing permission rules pointed out that people were using WEBGUIDE primarily as a threaded discussion medium and rarely as a knowledge construction space. Furthermore, their ability to construct shared group perspectives on discussion topics was severely hampered by the lack of support for negotiation in the system.

6. *Distinguishing the system's capabilities*.

In iterating the design of WEBGUIDE it became increasingly clear that what the system "wanted to be" was a *medium for construction of knowledge*. Yet, users were more familiar with discussion forums and tended to ignore the perspectives apparatus in favor of engaging in threaded discussion. These are very different kinds of tasks: collaborative knowledge construction generally requires a prolonged process of brainstorming alternative ideas, working out the implications of different options and negotiating conclusions; discussion can be much more spontaneous.

This suggests that more clarity is needed on the question: what is the *task*? If people are going to use WEBGUIDE for collaborative knowledge construction then they need to have a clear sense of pursuing a knowledge construction task. The Logan students have such a task in articulating positions on acid mine drainage. However, much of their knowledge construction takes place in classroom discussion. They use WEBGUIDE largely as a repository for their ideas. The seminar has been concerned with understanding a series of readings, so its participants have been more interested in exchanging isolated questions or reactions than in formulating larger integrative positions. For the remainder of the seminar, we will be trying to develop ideas for a collaborative paper on the nature of computer collaboration. This may provide the kind of focused task needed to exercise more of WEBGUIDE's potential.

Our experience to date already suggests the complexity of trying to support collaborative learning. We should probably distinguish within the software interface functions that support discussion from those that support knowledge construction. But this should be done in such a way that spontaneously discussed ideas can later be readily integrated into longer-term knowledge construction processes. Similarly, additional functionality – most notably support for group negotiation – must be added, differentiated and integrated. New capabilities and uses of WEBGUIDE can increase its value, as long as confusions and conflicts are not introduced. For instance, providing facilities for people to maintain lists of annotated Web bookmarks, things-to-do, favorite references, up-coming deadlines, etc. within their personal perspectives might not only give them familiarity with using the system, but would also build toward that critical mass of usage necessary for meaningful adoption.

It has become a cliché that computer mediation has the potential to revolutionize communication just like the printing press did long ago. But the real lesson in this analogy is that widespread literacy involved slow changes in skills and practices to take advantage of the technological affordances. In fact, the transition from orality to literacy involved a radical change in how the world thinks and works (Ong, 1998). Although social as well as technical changes can be propagated much faster now, it is still necessary to evolve suitable mixes of practices and systems to support the move from predominantly individual construction of knowledge to a new level of *collaborative cognition*.

Our investigation of the above six issues will guide the next stage of our on-going exploration of the potentials and barriers of computer mediated collaborative learning on the Web with perspectives. Because we expect the exploration of computer mediated collaborative learning to be a termless process, we will stop this paper here without a conclusions section making final claims.

ACKNOWLEDGMENTS

The WEBGUIDE research is a collaboration of the author with Rogerio dePaula and other L^{3}D members, Ted Habermann and his group at NOAA, Dan Kowal and his middle school students, the participants in the WEBGUIDE seminar, Thomas Herrmann and his students at Dortmund, and the researchers in the ICS "Articulate Learners" project. The work reported here was supported in part by grants from NSF IRI-9711951, the McDonnell Foundation and NSF EAR-9870934.

REFERENCES

[*Note*: The sculptures decorating this paper are documented at:

http://www.cs.colorado.edu/~gerry/personal/recreation/form/.

Publications authored by Stahl are available at:

http://www.cs.colorado.edu/~gerry/publications/]

Boland, R. J. & Tenkasi, R. V. (1995) Perspective making and perspective taking in communities of knowing, *Organization Science,* 6(4), pp. 350-372.

Bourdieu, P. (1972) *Outline of a Theory of Practice,* Cambridge University Press, Cambridge, UK.

Bruner, J. (1990) *Acts of Meaning,* Harvard University Press, Cambridge, MA.

Bush, V. (1945) As we may think, *Atlantic Monthly,* 176(1), pp. 101-108.

Caron, J. (1998) Wide area collaboration: A proposed application [submitted], *Computer Supported Collaborative Work (CSCW 98)*.

Cole, M. (1996) *Cultural Psychology,* Harvard University Press, Cambridge, MA.

dePaula, R. (1998) *Computer Support for Collaborative Learning: Understanding Practices and Technology Adoption*, Masters Thesis, Telecommunications Department, University of Colorado, Boulder, CO.

Habermas, J. (1971) Labor and interaction: Remarks on Hegel's Jena philosophy of mind. In *Theory and Practice,* Beacon Press, Boston, MA, pp. 142-169.

Hegel, G. W. F. (1807/1967) *Phenomenology of Spirit,* (J. B. Baillie, Trans.), Harper & Row, New York, NY.

Hewitt, J., Scardamalia, M., & Webb, J. (1998) *Situative design issues for interactive learning environments,* at http://csile.oise.on.ca/abstracts/situ_design.

Koschmann, T., Ostwald, J., & Stahl, G. (1998) Shouldn't we really be studying practice? [panel position paper], *Computer Supported Cooperative Work (CSCW 98)*.

Koschmann, T. & Stahl, G. (1998) Learning issues in problem-based learning: Situating collaborative information [workshop position paper], *International Conference on the Learning Sciences (ICLS '98)*.

Koyeve, A. (1947/1969) *Introduction to the Reading of Hegel,* (James Nichols, Jr., Trans.), Basic Books, New York, NY.

Lave, J. (1991) Situating learning in communities of practice. In L. Resnick, J. Levine, & S. Teasley (Eds.), *Perspectives on Socially Shared Cognition,* APA, Washington, DC, pp. 63-83.

Lave, J. (1996) Teaching, as learning, in practice, *Mind, Culture, and Activity,* 3(3), pp. 149-164.

Lave, J. & Wenger, E. (1991) *Situated Learning: Legitimate Peripheral Participation,* Cambridge University Press, Cambridge, UK.

Marx, K. (1844/1967) Alienated labor. In L. G. K. Easton (Ed.) *Writings of the Young Marx on Philosophy and Society,* Doubleday, New York, NY, pp. 287-300.

Marx, K. (1845/1967) Theses on Feuerbach. In L. G. K. Easton (Ed.) *Writings of the Young Marx on Philosophy and Society,* Doubleday, New York, NY, pp. 400-401.

Marx, K. (1867/1976) *Capital, Volume I,* (Ben Fowkes, Trans.), Penguin, New York, NY.

Maturana, H. R. & Varela, F. J. (1987) The Tree of Knowledge: The Biological Roots of Human Understanding, Shambhala, Boston, MA.

McCall, R., Bennett, P., d'Oronzio, P., Ostwald, J., Shipman, F., & Wallace, N. (1990) PHIDIAS: Integrating CAD graphics into dynamic hypertext, *European Conference on Hypertext (ECHT '90)*, pp. 152-165.

Mittal, S., Bobrow, D., & Kahn, K. (1986) Virtual copies at the boundary between classes and instances, *Object-Oriented Programming Systems, Languages and Applications (OOPSLA '86)*, pp. 159-166.

Nelson, T. (1981) *Literary Machines,* Mindful Press, New York, NY.

Ong, W. (1998) Orality and Literacy: The Technologizing of the World, Routledge, New York, NY.

Scardamalia, M. & 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, pp. 37-68.

Scardamalia, M. & Bereiter, C. (1996) Computer support for knowledge-building communities, *CSCL: Computer Supported Collaborative Learning*, pp. 249-268.

Schön, D. A. (1983) The Reflective Practitioner: How Professionals Think in Action, Basic Books, New York, NY.

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, *Proceedings of the Cognitive Science Society (CogSci '93)*, pp. 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. (1996) *Personalizing the Web,* Technical Report No. CU-CS-836-96, Department of Computer Science, University of Colorado, Boulder, CO.

Stahl, G. (1999) Collaborative information environments to support knowledge construction by communities [submitted], *AI & Society*.

Stahl, G., Sumner, T., & Owen, R. (1995a) Share globally, adapt locally: Software to create and distribute student-centered curriculum, *Computers and Education. Special Issue on Education and the Internet,* 24(3), pp. 237-246.

Stahl, G., Sumner, T., & Repenning, A. (1995b) Internet repositories for collaborative learning: Supporting both students and teachers, *Proceedings of Computer Support for Collaborative Learning (CSCL '95)*, pp. 321-328.

Tomasello, M., Kruger, A. C., & Ratner, H. (1993) Cultural learning, *Behavioral and Brain Sciences*, pp. 495-552.

Vygotsky, L. (1930/1978) *Mind in Society,* Harvard University Press, Cambridge, MA.

PERSPECTIVES ON COLLABORATIVE KNOWLEDGE-BUILDING

PROJECT SUMMARY

Knowledge-Building: This project proposes to develop an approach to information management grounded in an innovative theory of *learning and collaboration as knowledge-building*. The theory is oriented toward guiding development of technology that can better fulfill growing societal requirements for Web-based support of groups. It offers an analysis of collaborative knowledge-building activities and of Knowledge-Building Environments (KBEs) to support these activities. To do this, it brings together and synthesizes approaches and concepts from situated learning, activity theory, hermeneutics, distributed cognition, and related theories from philosophy, social science, education, and computer science.

Knowledge-Building Environments: The proposed work centers on a new way of using computers to support collaboration by *integrating information management support for various activities of knowledge-building*. Such KBE software prototypes go beyond superficial discussion or chat and beyond choices and decisions among fixed options to support the co-construction of deep knowledge, innovation, and shared understanding. They support learning, working, and innovation over time within groups that may be physically distributed.

Perspectives: A key innovation for KBEs in this project is the modeling of the *interplay between individuals and the group* in collaborative settings. This is done by providing personal and group computational perspectives: virtual workspaces whose contents are interrelated by automated inheritance mechanisms. Computational perspectives provide a new, dynamic, personalized form of on-line information management that supports the fundamental structure of collaboration. They help to manage a shared information space so that participants view information relevant to themselves and can process (edit, rearrange, reconceptualize) that information without affecting anyone else's personal perspective. Then, through supported negotiation activities, information is migrated to sub-group and group perspectives, where it represents shared knowledge.

Research and Training Focus: The project will directly employ four students and will involve many more in seminars and class projects. Because KBE software is essentially a new form of learning technology, education students as well as computer science and other students will be involved in designing, developing, deploying, and assessing the software. This will help to create an *interdisciplinary educational technology research focus* that the PI and his colleagues have already begun to foster.

Software Development: The proposed work will build an infrastructure for local, national, and international collaboration on KBE software development, providing a more reliable *basis for assembling KBE prototypes* customized to particular deployment sites. An open source perspectives server will be released, allowing researchers to develop KBE interface components that simply call this server for database access and dynamic computation of perspective contents. A standard for data interchange with the server and for interoperability among KBE systems will facilitate a component architecture and the use of shared tools to assess KBE usage.

Study Sites: The project will assess KBEs with perspectives in realistic study sites: an academic research group, a collaborative learning seminar, a corporate training setting, and an industrial design group. *Quantitative analysis* of captured textual contents will be compared with results from non-integrated threaded discussion systems and other groupware. *Qualitative analysis* of surveys and field notes will investigate issues of deployment, adoption, social practice, utility, and effectiveness.

Project Impacts: The proposed project should result in progress in the development and assessment of KBEs, an emerging form of software with a potential to significantly extend human cognition by supporting collaborative knowledge-building activities and by providing persistent external memory of what took place during the collaboration. The release of a perspectives server with its associated standards will provide a concrete basis for catalyzing *local, national, and international collaboration* among KBE researchers. This will promote KBE research as an important new research focus.

PROJECT DESCRIPTION

This project is guided by a *theory of collaborative Knowledge-Building Environments* (KBEs) that we are developing. This theory proposes the following principles:

- *Collaborative knowledge-building* is a particular view of group learning that focuses on a range of activities that take place within communities, as opposed to focusing on learning as the transmission of bits of information to individual learners.
- Collaborative knowledge-building takes place largely through the interaction among people with different understandings from *multiple personal and group perspectives*.
- Such knowledge-building within groups can be helped by appropriately designed *information technology* (IT) that supports various knowledge-building activities and supports interaction among alternative perspectives.

The form of IT that we are interested in – collaborative Knowledge-Building Environments – represents a distinctive approach that overlaps related work in Computer-Supported Collaborative Learning (CSCL) and Computer-Supported Cooperative Work (CSCW). IT support for learning is traditionally oriented toward the transmission of information to individual students. Even where it is based on a view of student construction of knowledge, as with Intelligent Tutoring Systems (ITS) for algebra or physics, the goal is measured by testing the incorporation of pre-defined content or methods into the individual's understanding (Wenger, 1987). A more student-centered, constructivist approach is taken by Interactive Learning Environments (ILE), which might, for instance, allow students to create ecologies in `SimLife` to learn biology, or programs in `Turtle Logo` to explore math concepts (Papert, 1980). In contrast, a KBE primarily supports the group process and leaves matters of content up to the participants (which may include a teacher who raises particular content issues and helps maintain focus). In this way, it applies CSCW approaches to CSCL. A review of CSCW technology for groups (Kraemer & Pinsonneault, 1990) distinguishes group communication support systems (GCSSs) from decision support (GDSSs). GCSSs are specific communication media like email and video-conferencing. In providing computational tools for group decision making, GDSSs tend to support isolated, focused activities that integrate products of individual work. In contrast, a KBE aims to support a broad spectrum of knowledge-building activities – both individual and group – in a more seamless fashion. It supports the construction of areas of knowledge through group inquiry over extended periods of time. It also supports the interplay of individual and group more comprehensively, through integrated mechanisms of "computational perspectives" and negotiation that treat the group as more than just the sum of the individuals.

Assessments of CSCL and CSCW systems have defined a number of key issues for evaluating the problems and successes of such systems. For instance, in simple threaded discussion forums common problems include: short threads (a tendency for discussions to die quickly), low participation (lack of motivation to participate), few cross-references (little convergence of ideas), and superficial content (minimal depth of investigation) (dePaula, 1998; Guzdial & Turns, 2000; Hewitt & Teplovs, 1999). On the other hand, GDSSs and GCSSs attempt to decrease communication barriers within the group, while increasing task-oriented focus, depth of analysis, and decision quality (Connolly, 1997; Kraemer & Pinsonneault, 1990). Social informatics studies have raised additional issues of software deployment and adoption in addition to questions of usability and utility (Kling, 1999). These are some of the dimensions along which KBEs must be assessed within realistic learning and working social contexts.

To date, the PI and his colleagues have begun to develop KBE theory in conjunction with Web-based KBE prototypes that support many of the activities described in the theory and that have been tested informally in collaborative learning classrooms. In particular, computational support for *personal and group perspectives* has been developed and tried out. Support for computational perspectives was explored in the PI's dissertation (Stahl, 1993a) and has since been refined and adapted to the Web (Stahl, 1999a). This work has been described in relevant CSCL and CSCW conferences (see references by Stahl). The proposed project will build on existing concepts and prototypes, extending them substantially by: implementing a technical infrastructure to support data interoperability, integration of functionalities, and rapid prototyping; deploying customized KBEs in specific study sites; observing the social impacts of these IT systems in the work settings; revising the theory based on empirical findings; and fostering a community of researchers working on IT support for knowledge-building in workgroups.

1. Preparation for Proposed Work under Prior NSF Support

1.1. Organizational Memory and Organizational Learning (CSS)

"Conceptual Frameworks and Computational Support for Organizational Memories and Organizational Learning (OMOL)," PIs: Gerhard Fischer, Gerry Stahl, Jonathan Ostwald, September 1997 – August 2000, $725,000, from NSF CSS Program #IRR-9711951.

This grant prepared much of the background for the proposed work. The OMOL project started from a model of computer support for organizations as Domain-Oriented Design Environments (DODEs) in which both domain knowledge and local knowledge are stored in the form of artifact designs and associated design rationale (Fischer, 1994). This CSCW model evolved into one of Collaborative Information Environments (CIEs), that emphasized the interactive, asynchronous, persistent discussion of concepts and issues within an organization (Stahl, 1998; Stahl, 2000a). Gradually, interest in organizational learning aspects led to involvement in CSCL and the model of collaborative Knowledge-Building Environments (KBEs) (Fischer et al., 1999). A number of software prototypes were developed to explore the use of the Web as a communication and collaboration medium. Of these, the most important for the proposed work are the following:

- `DynaClass`: A discussion forum for use in college courses. It features ties to `DynaGloss` and `Sources` as well as email notification and specialized displays (Ostwald, 1999).
- `WebGuide`: Differs from `DynaClass` in providing more control over rearrangement of notes; features computational Perspectives (Stahl & dePaula, 1998; Stahl et al., 1998).
- `DynaGloss`: A system for defining technical terms and keywords and for debating the definitions and reviewing the history of debate; linked to `DynaClass` and `Sources` in that each term shows all the locations in these other systems where the term is explicitly referenced.
- `Sources`: A system for annotating bibliographical entries; uses terms from `DynaGloss` as keywords.
- `InfoMap`: An interface component for creating a graphical display of linked notes like a threaded discussion; providing convenient drag-and-drop functionality.

Work on this grant led to the focus on KBEs as models of computer support for organizational memory and organizational learning. In particular, it provided a number of different systems, each with useful functionality, and brought home the need to define component standards so the functionalities can be combined more flexibly. As we tested and deployed these systems, we confronted serious issues of adoption and focused our concerns increasingly on socio-technical and social informatics (Kling, 1999) issues: motivation, media competition, critical mass, social practices, seeding, management, re-seeding, convergence of ideas, peer-to-peer collaboration, deployment strategies. These issues led to a new research agenda (Stahl, 1999b) and this proposal.

1.2. WebGuide and Environmental Perspectives (NOAA)

"Collaborative Web-Based Tools for Learning to Integrate Scientific Results into Social Policy," PIs: Ray Habermann, Gerry Stahl, November 1998 – July 1999, $89,338, NSF, #EAR-9870934.

This grant funded the initial implementation of `WebGuide` as an integrated `Java` applet KBE supporting personal and group Perspectives. It was a joint effort between the PI, a middle school teacher, and a research group at the National Oceanographic and Atmospheric Administration (NOAA) labs in Boulder. The teacher taught an environmental science class in which he wanted to spend the year having his students interview various adults and construct a set of contrasting perspectives (conservationist, regulatory, business, community) on a particular local environmental issue that the students had previously been involved in. `WebGuide` was used by the students to collect notes on their interviews and to formulate personal and team perspectives on the issue. Results of this software trial were analyzed and presented at conferences (Stahl, 1999a; 1999b; 1999c; Stahl & Herrmann, 1999). These findings led to a number of revisions of `WebGuide`, including the separation of the Perspectives mechanism from the Web interface, and recognition of the need for software architectures, standards, and components to support flexible rapid prototyping of KBEs.

1.3. Collaboration in KBEs (CILT)

"Interoperability Among Knowledge-Building Environments," PI: Gerry Stahl, September 1999 – August 2000, $9,124.21, from NSF-funded Center for Innovative Learning Technology (CILT), Subcontract #17-000359 under NSF grant #EIA-9720384.

This is a current seed grant whose purpose is to stimulate collaboration among KBE research groups. Part of the intention of the grant was to prepare a proposal for fuller funding, such as the present proposal and its currently pending complementary NSF proposals for "IT Support for Knowledge-Building in Workgroups" and "ITR/EWF: Collaborative Research on Knowledge-Building Environments: Growing a National and International Research Community for Distance Learning Information Technology." This grant has already resulted in a semester-long student project involving three graduate and three undergraduate students (one collaborating virtually from Germany) creating an XML DTD that defines a data format for data imported from several different KBE prototypes and displayed in a Web browser using XSL. The grant supported a workshop entitled "Collaborating on the Design and Assessment of KBEs in the 2000's" at CSCL '99 at Stanford. This workshop attracted over 60 participants and was preceded by an on-line discussion of 28 submitted position papers. This grant has led to the emphasis on collaboration among KBE research groups and the need to put into place some of the technical and social conditions for such collaboration (Stahl, 1999a), as proposed here.

2. Overview of Proposed Work

2.1. Theory of Collaborative Knowledge-Building Environments

Collaborative Knowledge-Building

Information Technology (IT) is a broad field that can be conceptualized in various ways. Traditionally, the computer was thought of as a medium for storing and delivering data, that can then be used by people in their work. More recently, the computer (especially with the Web) has become a medium of communication, through which people share information and knowledge. This communication can take a variety of forms. In simple forms of e-commerce or on-line voting, people submit their decisions about a fixed list of choices. In chat and most email, people exchange greetings and opinions, generally without changing those opinions. Many systems in recent years have tried to support a particular form of communication or social interaction like brainstorming or decision-making – often with very positive results (Connolly, 1997; Vogel et al., 1987). We are interested in a distinct but broader process of communication which we term *collaborative knowledge-building*. Here, groups of people construct new knowledge through interaction of their ideas and perspectives, usually eventually preserved in documents or other artifacts.

Our theory of collaborative knowledge-building (Fischer et al., 1993/1998; Stahl, 1975; 1993a; 1993b; 1999c; 2000a; 2000b; Stahl & Herrmann, 1999) proposes a concept we call *the synergistic moment*; we intend to investigate the validity of this concept in the proposed project. The synergistic moment is the critical point during collaboration in which a group constructs meaning that transcends what any participant may have "in mind." The *shared understanding* that is generated in this process is a subtle phenomenon: It does not mean that everyone is in complete agreement or even that each individual has the same internal cognitive representations of what is discussed. Rather, it means that a certain group view has been expressed. The unit of analysis for describing this is the group, and is manifested in the group's discourse. Individuals may agree to disagree with the group understanding, and careful investigation may reveal that individual understandings differ from the group's view (Hatano & Inagaki, 1991). The intersubjective "sharing" is not a correspondence or overlapping of individuals' mental content, but a coordination or interaction of their participation in joint socio-cultural activity (Matusov, 1996).The synergistic moment is an emergent property of the group dialog as a cacophony of voices (Bakhtin, 1986). It could easily pass unnoticed as a magical fount of creativity; to more deeply understand it likely requires "thick description" (Geertz, 1973) and detailed interaction/discourse analysis (Jordan & Henderson, 1995), and therefore presupposes that the interaction was captured in some medium. Fortunately, the literature on CSCL contains a number of incisive analyses (Roschelle, 1998) of the synergistic moment, although they do not highlight it as such.

The synergistic moment is a result of perspective-sharing (Boland & Tenkasi, 1995), but at the group rather than the individual level. It overcomes the problem pointed out by Feltovich et al. (Feltovich et al., 1996), that any one perspective may limit the ability to comprehend creatively the complexity of a topic under discussion. What typically happens is that one person makes a statement from her personal perspective; someone else interprets that statement from his own perspective and responds accordingly; others continue this process so that the discourse consists implicitly of reinterpretations from various perspectives. The drive to establish intersubjectivity and shared knowledge is powered by socio-cognitive conflict and contention among perspectives according to studies by Piaget and his followers (Perret-Clermont & Schubauer-Leoni, 1981). The dialog proceeds through sequential turn-taking and attempts to repair "misunderstandings" as understood from particular perspectives and reinterpreted from others. Thanks to the human drive to impose coherent social meaning structures (Geertz, 1973), a synergistic group understanding emerges. This shared understanding can play a central role in the further activity of the group and can be more or less adopted by individuals into their personal perspectives. Although the synergistic moment seems to the participants to emerge spontaneously, it can be understood as the result of many identifiable knowledge-building activities, as represented in our model (below).

Perspectives in Knowledge-Building

According to hermeneutics – the philosophy of interpretation – human understanding is fundamentally perspectival. We construct knowledge from our situated perspective in the world: our historical position, cultural tools, and practical interests (Gadamer, 1960/1988; Heidegger, 1927/1996; Stahl, 1975). Computational support for knowledge-building can represent our interpretive perspectives with computational Perspectives (Boland & Tenkasi, 1995; Nygaard & Sørgaard, 1987; Winograd & Flores, 1986). (In this proposal, Perspective–with-a-capital-P will refer to the proposed computational mechanism that mirrors human interpretive perspectives-with-a-lower-case-p.) In this sense, Knowledge-Building Environments (KBEs) with computational Perspectives are designed to support the essential structure of collaboration. A key hypothesis of the proposed work is that KBEs benefit from an approach that represents the perspectival nature of collaboration. A goal of the project is to facilitate the incorporation of a computational Perspectives mechanism in KBEs – both in our own prototypes and in the work of other KBE research groups around the world.

Computational Perspectives have been explored by the PI in a number of software prototypes, in his dissertation system, and in his theoretical publications (Stahl, 1993a; 1993b; 1995; 1998; Stahl & Herrmann, 1999; Stahl et al., 1995). In a single-user system, computational Perspectives may correspond to different domains or professional viewpoints on a design problem, such as electrical, plumbing, structural, and heating concerns in architecture (Fischer et al., 1993; 1993/1998). In a KBE to support collaboration, computational Perspectives typically provide personal or group workspaces for the development of different sets of ideas. In this way, they can model the relationships among the various personal and group interpretive perspectives at work in the construction of collaborative knowledge.

We hypothesize that computational Perspectives can support the synergistic moment in collaborative knowledge-building by providing the necessary contact among different personal Perspectives, allowing them to interact, and then locating the results in a group Perspective. By situating the traditionally ephemeral synergistic moment within an explicit structure of computational Perspectives and by doing so in a persistent way, a KBE provides new opportunities for group self-reflection.

An important complement to Perspectives is *negotiation*. Negotiation is a process through which divergent personal perspectives converge on a collaborative shared understanding. When Perspectives and negotiation are effectively "intertwined" in a KBE, they compensate for each other's potential problems: Negotiation converges ideas so that everyone can benefit from the ideas of other perspectives, while personal Perspectives allow people to work on their own views while potentially time-consuming negotiations are underway (Stahl & Herrmann, 1999).

For instance, when `WebGuide` – a KBE with computational Perspectives implemented by the PI – was used in a middle school environmental science classroom, students each had their own personal Perspective in which to develop their own responses to questions posed by the teacher. The teacher's questions to the whole class were posed in the class' group Perspective. From there they were automatically inherited into the team Perspectives. The content of the team Perspectives was, in turn, inherited into the personal Perspectives of team members. Gradually, students migrated their ideas to team Perspectives that represented either conservationist, governmental, corporate, or citizen perspectives on the ecological controversy – depending on which perspective team the student was part of. Then they could work with the ideas of their team-mates and negotiate their team position. In the end, the different teams negotiated to spell out agreements and disagreements (Stahl, 1999c).

The analysis of the synergistic moment suggests that negotiation need not take the explicit, rationalist forms typical of GDSSs, such as voting. Group results may emerge naturally out of the intertwining of Perspectives in group discussion. A challenge of the proposed work will be to develop software support for capturing such results and migrating them un-intrusively to group Perspectives.

The Potential of IT Support for Knowledge-Building

IT support has the potential of transforming the activities underlying the synergistic moment. For one thing, it would make those activities publicly accessible. The group could then reflect upon the emergence of its shared understanding by looking over the persistent record of its dialog. Such reflection might prove especially useful in contentious situations or for newcomers who were not part of the original dialog and are motivated to re-open the issue – as illustrated by Matusov (1996). Furthermore, computer support of perspectives could make explicit the interplay of different personal Perspectives and the migration of ideas and their interpretations between personal and group Perspectives. Ironically, perhaps, the “asynchronous” medium of the Web would allow group members to interact simultaneously – without waiting for sequential turns – thereby overcoming what Peters (1998) characterizes as “the hardest argument against democracy: the ability of only one person to speak and be heard at a time” (p. 261). Of course, as we have already discovered with the Web in general, the increased flood of ideas raises complex information management issues. We do not yet understand the full social impact of the envisioned KBEs – we will only know how they are used once they have been implemented, deployed in naturalistic settings, and observed.

Theories of human cognitive development emphasize the important role of external memories to extend short-term and long-term human memory (Donald, 1991; Norman, 1993). They also stress that individual cognition is a social product, highly mediated by social symbol systems, cultural artifacts, processes of structuration, and group collaboration (Bourdieu, 1972/1995; Geertz, 1973; Giddens, 1984; Hutchins, 1996; Vygotsky, 1930/1978). This suggests that computer support for collaboration has the potential to significantly advance the power of human cognition. In addition to maintaining a persistent external memory, IT can help people to be more reflective and creative – as has been demonstrated in computer support for brainstorming and decision-making (Connolly, 1997; Vogel et al., 1987). However, as our research to date indicates, despite the fact that the Web seems to offer a promising technological base for such a development, computer-supported collaboration is a complex process that requires a sophisticated body of knowledge that we are just beginning to assemble. Moreover, the potential is beyond the reach of any single research group.

We believe that IT support for collaborative knowledge-building has not yet been developed to near its potential. KBE research has been carried on for over a decade now, starting with the `CSILE` system and continuing with `KIE`, `CoVis`, etc. (Cuthbert, 1999; Pea, 1993; Scardamalia & Bereiter, 1991; 1996). Recently, commercial systems like `KnowledgeForum`, `WebCT`, and `LearningSpace` are catching on. However, as yet there has been no systematic attempt to support the variety of activities that are involved in knowledge-building. There is no general theory of collaborative knowledge-building as a social process. Existing research tends to target specific contexts like middle school science with specialized closed systems, rather than developing interchangeable, open source components that can be applied in a full range of contexts. While networks of KBE researchers are coming together in other countries, there is little organized effort to collaborate in the US. The proposed project aims to change this situation. However, collaboration across institutions cannot be started by just wishing for it. This project tries to put some of the necessary conditions in place by developing technical infrastructure (standards, a Perspectives Server, interface components) and initial results that can be used to stimulate discussion and collaboration among KBE researchers locally, nationally, and internationally. Interoperability and collaboration will allow isolated advances to be exchanged, new functionalities to be shared, and test data to be compared.

A Model of Collaborative Knowledge-Building

One approach to better understanding how to design computer support for collaborative knowledge-building in social settings is to conceptualize the various constituent activities involved in individual and social knowledge-building. The diagram below from (Stahl, 2000b) provides a starting point for this, combining aspects of activity theory, situated learning, hermeneutic philosophy, and distributed cognition theory (Chaiklin & Lave, 1993; Cole, 1996; Engeström et al., 1999; Gadamer, 1960/1988; Hutchins, 1996; Lave & Wenger, 1991; Nardi, 1996).

The idea of this diagram is that knowledge-building can proceed through many different activities. The sequential structure of the model is only illustrative of an ideal conceptualization. We understand that these activities complexly overlap in practice. The possible relationships among the individual activities – and particularly the interactions between the personal and social – can be complex and varied. The purpose of the diagram is to suggest a number of distinct activities that could be supported by a KBE with multiple functionality. The sequential labeling of these activities corresponds to proposed KBE components listed in Table 1 below, and it is not intended to imply a necessary order to the activities.

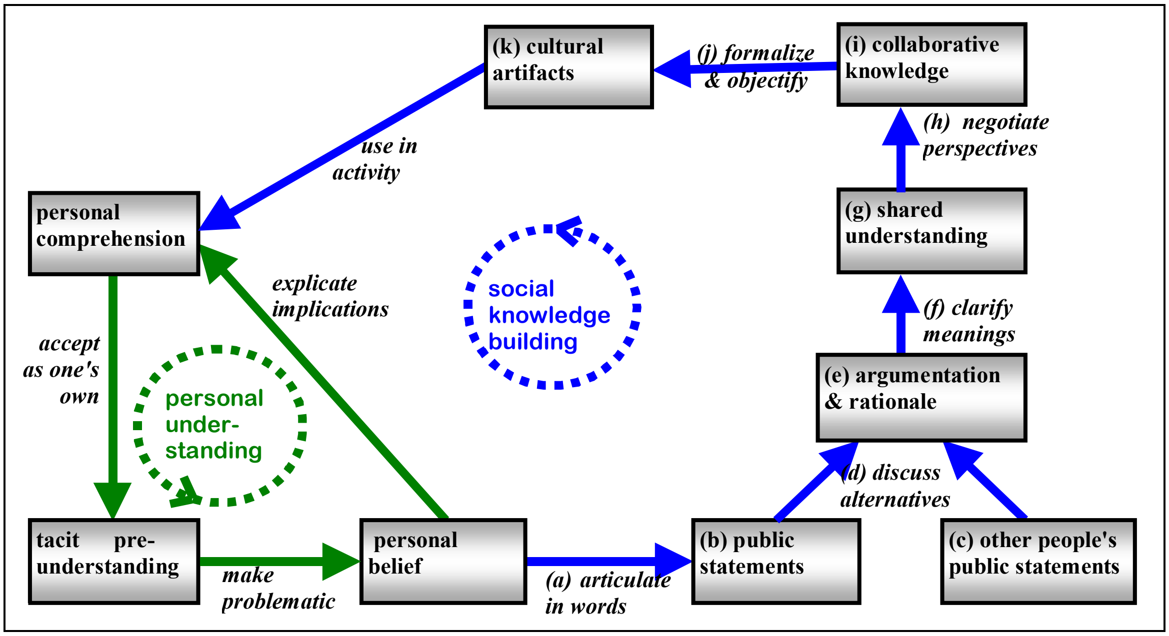


Figure 1. A model of personal understanding and social knowledge-building.

A set of seminal books and articles in Computer-Supported Collaborative Learning (CSCL) has formulated a view of *learning* as a social process of collaborative knowledge-building within communities of practice (Brown & Campione, 1994; Brown & Duguid, 1991; Lave, 1991; Lave & Wenger, 1991; Pea, 1993; Scardamalia & Bereiter, 1996; Wenger, 1998). However, these texts do not make the set of cognitive and social activities that underlie such a view explicit in the manner attempted in our KBE theory.

Starting in the lower left corner, Figure 1 shows a cycle of personal understanding. The rest of the diagram depicts how personal beliefs can be articulated in language and become part of social interaction. Note that the results of social knowledge-building eventually feed into personal understanding, providing the evolving toolkit of culturally-based individual cognitive capabilities. The depicted knowledge-building activities are discussed briefly below in the context of proposed computer support.

IT Support for Knowledge-Building Activities

Each of the activities of social knowledge-building pictured in Figure 1 can be supported computationally. Table 1 lists an illustrative form of support for each. It also lists corresponding prototypes that we have developed. Support for each activity is briefly discussed following the table.

Table 1. Forms of computer support for knowledge building activities.

	Knowledge-building activities	Forms of computer support	Prototype systems
a	articulate in words	articulation editor	`DynaClass`
b	public statements	personal Perspective	`WebGuide`
c	other people's public statements	comparison Perspective	`WebGuide`
d	discuss alternatives	discussion forum	`DynaClass`
e	argumentation & rationale	argumentation graph	`InfoMap`

f	clarify meanings	glossary discussion	`DynaGloss`
g	shared understanding	glossary	`DynaGloss`
h	negotiate perspectives	negotiation support	`WebGuide`
i	collaborative knowledge	group Perspective	`WebGuide`
j	formalize and objectify	bibliography discussion	`Sources`
k	cultural artifacts and representations	bibliography or other community repository	`Sources`

a) Computer support should facilitate the process of articulating ideas and preserving them in convenient forms. Most KBEs, including discussion forums like `DynaClass`, provide an *editor* for articulating ideas. Some KBEs have tried to introduce procedural facilitation, scaffolding, or prompting to encourage someone to articulate an appropriate expression (Slotta & Linn, 2000). Other approaches would be to provide an outline editor or a brainstorming area.

b) Public statements by one person confront those of other people. Computer support can represent the different *perspectives* from which these statements emerge. Perspectives are more general than representations of individuals themselves, because one person can offer statements from multiple perspectives and several people can agree on a common perspective. Perspectives can be related to one another, for instance deriving from a common perspective that they share. Computational representations of perspectives in a KBE like `WebGuide` (Stahl, 2000a) make explicit the important relationships among personal and group perspectives, as well as providing means for individuals and collaborative teams to articulate their own perspectives.

c) A KBE with support for Perspectives should provide *comparison Perspectives*, in which one can view and contrast alternative Perspectives and adopt or adapt ideas from other people's Perspectives. Comparison Perspectives in `WebGuide` aggregate ideas from various individual and/or group Perspectives and allow for comparison of them (Boland & Tenkasi, 1995; Stahl, 1999c). Other systems like D^3E (Sumner & Buckingham Shum, 1998b) facilitate commentary on documents by other people, such as reviews of journal articles.

d) The most common element in current KBEs is the *discussion forum*. This is an asynchronous, interactive communication system like `DynaClass` that allows people to respond to notes posted by one another. Typically, there is a thread of responses to entered notes, with a tree of divergent opinions. A KBE should go beyond superficial undirected discussion to converge on shared understandings (dePaula, 1998; Guzdial & Turns, 2000; Hewitt & Teplovs, 1999).

e) Although every note in a discussion forum is a response to another note, the discussion may have a more complex implicit structure. One note might argue for or against another or provide evidence to back up the claim of another note, for instance. Such an argumentation structure can be made explicit and formalized in a representation of the *argumentation graph*. A component like `InfoMap` that displays the structure of notes graphically can contribute to participants' meta-level comprehension of their knowledge-building activity, pointing out where additional evidence is needed or where alternatives have not been explored (Buckingham Shum & Hammond, 1994; Donath et al., 1999; Suthers, 1999).

f) An important requirement for constructing group knowledge is the establishment of shared understanding. This can be fostered by clarifying the meaning of important terms used in various competing claims. A *glossary discussion* can make explicit how different participants understand the terms they use, as in `DynaGloss` or `DocReview` (Hendricksen, 1999).

g) The glossary discussion should result in a group *glossary* of the agreed upon definitions of important terms. Such a glossary already represents a form of group knowledge. The glossary is, of course, subject to future debate and emendation; it may make sense to define the glossary as a particular display of information from the glossary discussion (Stahl & Herrmann, 1999).

h) Perhaps the most delicate phase of knowledge-building is negotiation. Computer *support of negotiation* tends by nature to make explicit the factors entering into the negotiation process. This can be extremely harmful to the subtle processes of persuasion if not done sensitively. On the other hand, negotiation is critical to helping multiple perspectives to converge on shared knowledge. Computer support can provide a useful tool – as long as it is carefully integrated with other social activities that allow for implicit, culturally established interpersonal interactions (Stahl, 1999b). Group Decision Support Systems (GDSSs) have traditionally been independent systems, not integrated with the broader context of knowledge-building (Kraemer & Pinsonneault, 1990; Vogel et al., 1987).

i) The accumulation of negotiated shared knowledge results in the establishment of a *group perspective*. Like the alternative individual and team (or subgroup) perspectives, the group perspective may be represented in a KBE. In `WebGuide`, the content of the group Perspective is inherited into the individual and team Perspectives, because it has been accepted by the group. Individuals can then build on this shared knowledge within their own Perspective and even begin to critique it and start the whole cycle over (Stahl, 1999a).

j) Shared knowledge can be further formalized. It can be represented in another symbolic system or combined into a more comprehensive system of knowledge (Stahl, 1999c). For instance, in academic research knowledge is incorporated in new classroom lectures, conference presentations, journal articles, and books. The discussion of knowledge that has been compiled into publications can be carried out in a *bibliography discussion* component of a KBE.

k) Finally, representations of the new shared knowledge in publications and other cultural artifacts are themselves accepted as part of the established paradigm. Although still subject to occasional criticism, ideas in this form more generally provide part of the accepted base for building future knowledge. In academic circles, an annotated *bibliography* like `Sources` might provide a useful KBE component to support this knowledge building activity (Sumner & Buckingham Shum, 1998a).

A KBE goes beyond a single-purpose system – like a simple discussion forum – and supports more than one collaborative knowledge-building activity (Muukkonen et al., 1999). It retains a record of the knowledge that was incrementally collected – unlike common chat, newsgroup, and listserv systems that erase contributions after a short period of time. We hypothesize that it should help people to express their beliefs, to discuss them with others, to differentiate their own perspectives and adopt those of other people, clarify disagreements or misunderstandings, critique and explicate claims, negotiate shared understandings or agreements, and formulate knowledge in a lasting representation. Because KBEs are computational, they can provide facilities like searching, browsing, filtering, tailoring, and linking in order to group related ideas together automatically. KBEs can interface with other agents and software utilities – for instance sending emails to notify collaborators when important knowledge-building events have taken place (McLean, 1999). They can also dynamically format sets of notes in convenient displays for different purposes.

2.2. An Infrastructure for KBEs

Computational Perspectives

Computational Perspectives provide a new, dynamic, personalized form of on-line information management (Stahl, 1995). A Perspective defines an electronic workspace in which a person or group can develop ideas and manage information that belongs together – for instance because it represents the beliefs and viewpoint of a particular person, group, domain, or intellectual position. Perspectives structure a shared information space so that special coherent views can be built up and displayed. Although the mechanism of computational Perspectives is very general and flexible, the simplest way to use it in a small group is to define a personal Perspective for each member, one team Perspective for agreed upon ideas, and a comparison Perspective that collects the ideas from all the personal Perspectives.

The design philosophy behind computational Perspectives as implemented by the PI in `WebGuide` is that users have complete control over the content in their personal Perspectives. Thus, if my personal Perspective inherits conflicting ideas from different team Perspectives that I belong to, I can delete, edit, and rearrange those ideas at will. Other users can view the contents of my personal Perspective (except for content that I have designated as private) and they can copy items, link to them, initiate public discussions of them, and propose them for incorporation in team Perspectives – but none of this affects how the content of my Perspective is displayed to me. This allows me to build my own Perspective on the topics that are under consideration by the group. I can see what knowledge others are building, incorporate that knowledge into my Perspective, or join in with others to share, discuss, and negotiate. The same design philosophy applies of course to team Perspectives: team members jointly (through negotiation processes) have complete control over the content of their team Perspective.

Inheritance is a central defining mechanism of computational Perspectives as used in this proposal. The ability to define arbitrarily complex networks of Perspectives with multiple layers of sub-groups between the group Perspective and the individual personal Perspectives, and to have the automatic inheritance of content through the network distinguishes this approach from all other systems of "views" and "perspectives." Inheritance in this sense is not class inheritance, but "content inheritance." A given Perspective can inherit content from multiple other Perspectives. This content is aggregated (logical union) in the given Perspective, where it can be over-ridden with edits, deletions, rearrangements, and additions. The inheritance mechanism is derived from efficient approaches explored in hypermedia, including "delta memory" and "transclusion" (Boborow & Goldstein, 1980; McCall et al., 1990; Mittal et al., 1986; Nelson, 1981; Nelson, 1995). For a discussion of related work, see (Stahl & Herrmann, 1999).

Because new Perspectives can be defined (either in advance or during system use) to inherit from any (non-cyclical) other Perspectives, it is generally useful to define "comparison Perspectives" that aggregate the ideas from team members, including those ideas that have not been agreed upon and migrated to the team Perspective. This is handy for keeping an eye on what one's fellow team members are thinking. Typically, we have set up the inheritance network of Perspectives to have a diamond-shaped profile, diverging out from the total group Perspective via teams to all the personal Perspectives, and then converging back via team comparisons to the group comparison Perspective. This models a collaborative knowledge-building process that combines divergent brainstorming and convergent negotiation.

Functionally considered, a KBE with Perspectives like `WebGuide` consists of two primary subsystems: a Perspectival data selection computation and a set of interface displays of the selected data. When a display is requested, the system must search the database to determine which content notes should be displayed to the particular user in the requested Perspective. For instance, if I request to view your Perspective, the system must select notes that are defined within your Perspective or within any Perspective from which yours inherits (recursively), except for notes that are private or that have been over-ridden. Various special displays can also be computed using this inheritance computation by treating discussions, negotiations, historical archives, etc. as pseudo-Perspectives that have special inheritance and exclusion rules. Once the Perspectival data computation has been returned, the content can be displayed in specialized interfaces that provide different kinds of functionality useful for further knowledge-building.

An Open Source Perspectives Server

A specific task of the proposed project is to separate out the Perspective computation from `WebGuide` and structure it as a self-contained module with a well-defined application programming interface (API). This will form a Perspectives Server, a `Java` application that runs on the Web server along with the database system. It will be separate from the `WebGuide` client that will still run in a Web browser on the client's computer. This separation of functions into a server and a client will have many advantages. It will speed the functioning of `WebGuide` because the intensive computation of Perspective content will be done on a central server that is faster than typical student computers. Also, calls to the database system will take place locally rather than across the Internet. In terms of system development, it will mean that developers can build systems that incorporate Perspectives without having to worry about the Perspective algorithms or the database calls. They will use an API that lets them request data that should be shown to a given user in a given Perspective. They can then just focus on how best to display this data in the interface.

The Perspectives Server will be a self-contained Java application. It will be released as open source with clear documentation on how to use it to get Perspective data for display. The data will be delivered as an XML text stream that can be used by any Web technology, such as `HTML`, `Perl`, or `Java`. The data will be human-readable, making it easy for programmers to see what data is being passed. Although it is anticipated that the Perspectives Server will generally be used as a black box, its open source availability will allow programmers to modify it if necessary, such as to incorporate improvements to the XML DTD or in response to changes in Web technology. However, the Perspectives Server will be designed to make expansions of the database schema easy to incorporate without changes to the source code. This will allow new data structures corresponding to new multimedia data types.

The Perspectives Server will be a form of middle-ware, operating between the database and the client software. It will instantiate a three-tier, model-controller-view architecture that defines independent layers for the data schema or model, the data computation or control, and the interface display or view. The database management system can be any standard relational SQL system like `mySQL` or `Oracle`. The middle layer can be the Perspectives Server or a stripped down version that does not compute Perspectives. And the interface can be any kind of applet, Web page, or Web application that conforms to the API standard.

A Component Architecture for KBEs

The release of the open source Perspectives Server will not only facilitate the rapid prototyping of Perspectives-based systems for this project to use in its study sites, it will also allow other researchers to incorporate computational Perspectives in their KBEs. We have already had requests for this from researchers in California and in Germany.

The PI of this proposal is involved in several efforts to promote collaboration among KBE researchers. Among these, he is the PI on a seed grant from the NSF-funded Center for Innovative Learning Technologies (CILT) to foster data interoperability among KBE systems. Work on this is currently producing a draft XML DTD (eXtensible Markup Language – Document Type Definition) to provide a common data format that KBE data can be imported to and exported from. Data in this format can be displayed using CSS and XSL. Tools we are now developing will allow such data from any KBE to be analyzed with standard measures, allowing for instance the volume and characteristics of discussion threads on different systems to be quantified and compared.

This DTD will also provide the format for data transfer between the Perspectives Server and interface clients. Clearly, interface clients will have to be designed to accept and make use of data received in this XML format. An important aspect of the proposed work will be to define a set of standards: the XML DTD for data interchange, the API for the Perspectives Server, and the ability of interface clients to call the Server and make use of the data.

The PI will be communicating with other KBE researchers nationally and internationally to solicit their concerns about these standards and to work toward a consensus and adoption of a set of such standards. These standards do not have to be formally approved by international standards bodies; informal agreement within a set of collaborating research groups is all that is needed for substantial practical benefits. Use of these standards will allow for rapid prototyping and customization of systems for various study sites both within the proposed project and by other researchers. It will also move us significantly toward a future in which KBE components from different research groups can be intermixed so that new systems can take advantage of functionality developed at different sites. Certainly, every effort will be made to incorporate related international standards, such as those for XML, XLINK, and metadata.

Intelligent Hyper-Linking of Related Perspectives

System development efforts in Year I will focus on release of the Perspectives Server and in Year II on prototypes for the study sites. In Year III, we will enhance the power gained by organizing notes into personal and group Perspectives by adding the functionality to locate the notes most closely related to a given note, such as a new idea just entered into my personal Perspective or an old note proposed for inclusion in the group Perspective. Research in KBEs like CSILE has shown that it is difficult to locate related ideas within a shared database of discussion notes (Hewitt et al., 1998; Hewitt & Teplovs, 1999).

We will use Latent Semantic Analysis (LSA) (Landauer & Dumais, 1997; Landauer et al., 1998) to analyze the semantic content of notes and to measure the semantic relatedness of pairs of notes. LSA is based upon a statistical analysis (singular value decomposition) of co-occurrences of terms in a large corpus of text. It determines the relatedness of words even if they did not occur together explicitly – hence the term “latent”. LSA incorporates some refinements that make its performance closer to that of humans than similar methods (see special issues of *Discourse Analysis* 1997 and *Interactive Learning Environments* 2000 on LSA assessment studies). The PI and his graduate assistant are currently completing a four year project (sponsored by the McDonnell Foundation CSEP Program) that successfully uses LSA in a Web-based educational system tested in middle school classrooms (Kintsch et al., 2000; Stahl et al., 2000; Steinhart, 2000).

Automated linking of related notes will involve a fairly straight-forward application of LSA. It will be handled within the Perspectives Server, running on a computer with access to the necessary files for LSA. For each major application or study site, a corpus of domain-related text will be subjected to LSA analysis to define a semantic space. Periodically (e.g., each night) the site’s shared database of notes will be folded into this corpus to redefine the space and to compute the vector for each note within this space. In real time, when linking is requested for a new note, the note’s vector can be quickly computed and a list of notes with the closest vectors in the semantic space can be produced without noticeable delay.

We will experiment with different interfaces to try alternative approaches to incorporating this functionality into KBEs. For instance, it can be left to users to ask for lists of notes related to a given note. Alternatively, an agent can automatically check to see if there are notes within a given closeness to certain notes: newly entered notes, notes proposed for negotiation, notes being read or edited, etc. The agent can then suggest that links be established from the given note to similar ones. The different interfaces can be tried out in our study sites.

2.3. Study Sites for Evaluating Knowledge-Building in Workgroups

We will use local sites under our own control as alpha sites for testing our software, not only to eliminate bugs, but to try out different functionality and to refine the interface. More formal evaluation of the social impact of KBEs in workgroups will be conducted in corporate sites, primarily in the Boulder area. We have targeted two local situations (academic research and university learning) and two corporate sites (corporate training and industrial design).

Academic Research: the L^3D Center

The proposed work will take place within the Center for LifeLong Learning and Design (L^3D), a research group within the Department of Computer Science at the University of Colorado. We will try out our prototypes in a variety of applications within L^3D. Such self-application will give us first-hand experience with the requirements for the use of KBE software and with the practical problems of deployment and adoption. Increasingly, research at L^3D involves participants from different disciplines and even virtual subgroups, like colleagues at other universities and other countries. When, e.g., a research project involves participants from Boulder, Colorado, and from Dortmund, Germany, a Web-based collaboration medium is essential, and means for defining and negotiating personal, subgroup, and whole-group Perspectives seem particularly appropriate. Because users in this group are themselves software developers and researchers, they are particularly accepting of glitches and are reflective about design issues. This will provide a convenient and forgiving initial test site.

University Learning: College Seminars

Several members of L^3D, including the PI, offer undergraduate classes and graduate seminars. These courses typically emphasize student on-line discussion and group projects. They often have a content focus on the Internet. We have used `WebGuide` and other Web-based tools in past courses and will continue to use them in the future. We are exploring courses that not only span multiple disciplines, but also span the oceans. Again, here, Perspectives for subgroups (interest-, content-, discipline-, or location-based) make sense. Collaborative classrooms will provide a secondary test site in which KBE functionality and prototypes can be tried out and knowledge-building activities can be monitored.

Corporate Training: Athenaeum International DesignShops

We foresee our primary study site for this project being a corporate training setting, in which knowledge-building takes place under settings that may be advantageous for study. Athenaeum International (AI) is part of a distributed network of corporate training facilitators associated with MG Taylor. AI is located in Boulder and has established a good working relationship with the PI and with L^3D. AI specializes in the design and manufacture of custom movable furniture for rapid deployment at corporate training events, as well as the facilitation of such events. They are interested in incorporating knowledge-building software tools in their furniture to support the training process. (See attached letter of support from AI.)

A typical training event – or DesignShop™ – involves bringing together decision-makers from throughout a company to "reengineer" their corporation or re-think their high-level mission. This might involve a series of three-day workshops, or even an on-going sequence of quarterly gatherings. AI staff have noted a number of problems that they think could be addressed by innovative computer support: there is tremendous time pressure and everyone cannot express all their ideas and arguments; too much time is spent introducing materials; it is hard to retain important points and decisions; follow-through is tricky; documentation is labor-intensive. KBE support could include pre-workshop preparatory discussions, capturing of ideas that arise in the face-to-face meetings, organized documentation of debates that took place, and follow-up discussion, analysis, negotiation, decision-making, or follow-through.

AI would provide a challenging study site for KBEs. DesignShops are high-stakes events involving people who need to make efficient use of their time. The groups here would be larger than the test groups at the university, and would involve more intense face-to-face interaction. A workshop series is of limited duration, so success could be assessed quickly and changes made prior to a subsequent trial. Project staff would have the aid of AI's experienced group process facilitators to guide the design of the software support and of the deployment strategy, as well as to analyze the impact the software had on the social systems.

Industrial Design: Seagate Technical Design Centers

Another targeted corporate study site provides a rather different opportunity for investigating the use of KBEs. We have begun to investigate a particular work group within Seagate, a major hard disk manufacturer located near our university. Seagate is an established high-tech corporation. Through a series of acquisitions and mergers, it now consists of large design centers in Longmont (outside of Boulder, Colorado), Minneapolis (MN), Kansas City (IW), Redwood City (CA) and Singapore. Each of these design centers houses a few hundred employees working rather independently of the other centers.

We will focus on the effort of a Seagate Vice-President who is in charge of coordinating research on the problems of disk drive head tracking. In order to meet market demands that are projected as exceeding Moore's law – requiring a doubling of storage density every year – the designers who work on head tracking must solve complex issues in physics and mechanical engineering. Unfortunately, engineers at different centers work almost independently of each other, duplicating research and designing products that overlap in functionality and specifications. As a result of their different histories, the centers have very different cultures of work, interaction, and outlook. For instance, one center prides itself in minimizing costs while another spares little cost to make what they consider a quality product. Designers from the different centers are accustomed to different engineering paradigms and find it hard to talk to each other.

The research objective here would likely be to use KBEs to structure communication and collaborative design among the distributed design groups. There is already a shared Lotus `Notes` database in which everyone can view the specifications and deadlines for each group's product line. However, there is currently no medium of communication among the groups (other than generic email) and no persistent textual discussion of the posted data. This makes it hard to share interpretations, work on establishing common understandings, or collaborate on building knowledge from the data. There is no support for Perspectives that would represent the conflicting cultures of the distributed groups and allow for negotiation of these differences.

Other Potential Study Sites

We will try to work with both AI and Seagate initially to explore their suitability to this project. We may end up focusing on one or the other site for practical or theoretical reasons. It is always hard to predict how field research will develop over years, particularly in today's volatile marketplace, where key contacts change jobs and companies alter their strategies. For this reason, we have also established access to several other industrial sites similar to AI and Seagate: *StorageTech* is another major manufacturer of digital storage media; our contacts there are especially concerned with making their company a "learning organization" in which knowledge-building is recognized to be an essential aspect of work. *IBM* has a major support center near Boulder, with a help-desk organization of 700 employees who must continually collaborate to build knowledge of the products they support. L^3D has maintained a long-term relationship with two Japanese software companies, *SRA* and *PFU*, who are both interested in incorporating our ideas about lifelong learning and KBEs into their operations. These companies all have interesting settings where we would be welcome to deploy and observe our system prototypes if we have time during the proposed project.

Evaluation of Social Impact of KBEs at Study Sites

Evaluation will be conducted using converging methods to understand the complex, systemic issues around new technology deployment and use. Evaluation goals are two-fold:

- *Constructive*: We want to understand the environment with the objective of tailoring the design of the KBE technology to the study site, as well as constructively guide the deployment toward successful adoption.
- *Objective*: We want to objectively observe evolving use of the KBE technology; assess the nature of the IT impacts on coordination and collaboration; and evaluate the validity of the "collaborative knowledge building"

concept and applicability of the KBE theory; and refine the KBE theory based on results from observed practice.

Data collection will take the form of face-to-face in-depth interviews; phone and email-based "interviews" when subjects are at a distance (particularly in the case of Seagate); real-time non-participant observation in the workplace as well as via the KBE technology; surveys (particularly in the case of the AI training sessions, where there will be many more subjects); KBE database data collection; and document collection where appropriate.

In addition to coding and content analysis of field and interview notes, analytical approaches will also include structural analysis of discussion thread lengths and participation levels based on the KBE database data. We will perform discourse and content analysis of argumentative exchanges, and KBE-captured collaboration episodes.

Issues we will investigate include:

- What is the nature of collaborative knowledge-building, and what activities comprise it?
- How does asynchronous support for articulation affect participation with respect to a variety of factors including time, location, and social status?
- How does computational support for discussion, argumentation, and clarification affect consensus-building as well as conflict?
- Do subjects understand and put into practice the concept of computational Perspectives? How do subjects interact with and manage multiple Perspectives (personal, subgroup, group, comparison)?
- Can "synergistic moments" be captured? What portion of these moments occur on-line in KBE environments?
- What do subjects do with persistent dialog? Does persistence foster reflection on "synergistic moments," and under what circumstances (what shades of consensus and conflict)?
- What role do facilitators (in the case of Athenaeum International) and management play in the use of KBE technologies?
- What are the particular hurdles that must be overcome for KBEs to be useful in these particular organizational environments?

Since the evaluation is a multiple-person effort, and because there will likely be different students participating at different times during the pre- and post-deployment stage, we will carefully organize observation efforts, and make an effort to systemize field note format to the best possible extent. We will also conduct regular group data analysis meetings to coordinate the results of our efforts.

3. Objectives of Proposed Work

3.1. Objectives for Theory Development

- Investigate the phenomenon of the "synergistic moment" in interactions captured in the database.
- Investigate the utility and actual usage of computational Perspectives.
- Refine the model of collaborative knowledge-building activities based on project findings.
- Publish results of this project in a monograph.

3.2. Objectives for Technology Prototyping

- Define a standard for data interoperability among KBEs.
- Release open source import/export/display/analysis tools for KBE data interoperability.
- Release an open source Perspectives Server.
- Release example KBE interface components using different technologies.
- Develop an agent for intelligent hyper-linking of KBE notes.

3.3. Objectives for Deployment

- Deploy a KBE prototype in settings of academic research (e.g., L^3D research group).
- Deploy a KBE prototype in settings of university learning (e.g., seminars at the university).
- Deploy a KBE prototype in settings of corporate training (e.g., AI DesignShops).
- Deploy a KBE prototype in settings of industrial design (e.g., Seagate design groups).

3.4. Objectives for Evaluation

- Conduct initial evaluation of the way work is presently conducted among selected groups to: collect baseline data; carefully select groups to deploy to; and formulate deployment strategies.
- Assess existing challenges for communication and collaboration within and across groups through interviews and possibly email-based interview-type surveys for those participants at a distance.
- Observe deployment of KBEs and revise deployment strategies as necessary.
- Perform on-going qualitative (observation, interviews, and document collection) and quantitative (KBE database activity) data collection.
- Analyze collected data to evaluate the impact of the KBEs in the workgroups.

3.5. Objectives for Dissemination

- Build a local, interdisciplinary community of students and faculty to conduct KBE research.
- Collaborate with at least 3 US research groups and with at least 3 international KBE research networks.
- Maintain an active website with the results of this project, including the open source products.
- Present the work of this project in the Group ’01 and CSCL ’01 international conferences (to be held at the University of Colorado), and report the findings of this project at 3 or more other international conferences.

4. Plan of Proposed Work

4.1. Year I

Refine the model of collaborative knowledge-building activities through presentations to researchers and analysis of the "synergistic moment" in face-to-face collaboration.

Define a standard for data interoperability among KBEs using XML and XLINK for notes and relations among notes, including threaded discussions; explore the adequacy of this standard using local prototypes; circulate the proposed standard among other KBE researchers; integrate the proposed standard with emerging data standards.

Define a standard for KBE interface components including JavaBeans, using XML for data interchange; explore the adequacy of this standard using local prototypes; circulate the proposed standard among other KBE researchers; integrate the proposed standard with other emerging data standards.

Define a standard for KBE database servers to provide XML data structures to interface components meeting the above standards.

Release an open source Perspectives Server that efficiently computes data visible in a requested Perspective in the standard XML format. Provide open source and documentation so that other researchers can use this Server for Perspectives-based KBEs and can propose improvements to the Server.

Deploy a KBE prototype in a setting of academic research such as a local research group. The KBE will be used to discuss and design standards and software. Learn from this deployment experience.

Begin initial observation of two primary test sites (AI and Seagate) to prepare for eventual KBE deployment.

Collaborate with American research groups interested in KBEs (e.g., Berkeley, SIU, Stanford, Georgia Tech, Hawaii, SRI) and international KBE research networks (e.g., existing CSCL research networks in Germany, England, Norway, Finland, Canada) to coordinate software component design around agreed upon standards.

Establish a website to publicize the results of this project, including the open source products, and to provide an information center for collaborations. Incorporate a KBE in the website to foster interactive knowledge-building.

4.2. Year II

Release open source import/export/display/analysis tools for KBE data interoperability, using XML to define a DTD and using XSL and CSS to display the data. Document the standard and the tools on a website that makes them available to other researchers.

Release example KBE interface components in different technologies, including `Java`, `Perl`, and `HTML`. Make these available on a website with documentation and tutorials to help other researchers develop compatible components.

Deploy a KBE prototype in a setting of university learning such as a seminar offered by the PI, and observe use. The KBE will be used for students to develop, exchange, and negotiate reflections on shared readings and on collaborative writing projects.

Continue investigating the AI study site and assess the requirements for deployment of a KBE there.

Deploy a KBE prototype in a setting of corporate training such as a series of AI corporate training sessions. Commence post-deployment observation.

Present approach and intermediate findings of this project at the Group ’01 and CSCL ’01 international conferences. During Year II, these conferences will both be hosted at CU, with the PI serving as chair of the local arrangements committee at Group and chair of the program committee at CSCL.

4.3. Year III

Deploy a KBE prototype in a setting of industrial working such as a technical design group at Seagate, and observe and analyze use there. The AI deployment will also continue to be evaluated in Year III.

Develop an agent for intelligent hyper-linking of KBE notes using LSA technology to locate notes in a KBE database that are semantically related to a given note.

Continue on-going analyses of KBE use and examine findings with respect to the project's central hypotheses about KBEs. Publish a monograph on the project findings.

Report on this research at 3 or more international conferences.

5. Expected Impact of Proposed Work

The proposed work – with its development of theory, defining of architectures and standards, prototyping of systems, deployment at several study sites, evaluation of utility, and dissemination to collaborators around the world – may seem overly ambitious for a half-time (6 months per year for the PI) project. This would certainly be the case if a lot of the groundwork had not already been laid and if there was not an expectation that this project will take place within a context of supporting work. The PI has already published on KBE Theory and begun to sketch more in his seminars. Work is already well underway on separating the Perspectives Server out of `WebGuide` and defining an appropriate XML DTD for the data format – although these two projects have not yet been combined. Building a consensus on technical standards for KBEs has been initiated through the CILT grant and a popular workshop at CSCL ’99. It will continue at planned conferences, including Group ’01 and CSCL ’01 where the PI will be involved in planning. `WebGuide` and other L^3D prototypes have been used and evaluated in middle school and college courses, including the PI’s recent seminars. Colleagues in L^3D are developing systematic methodologies for analyzing and comparing discussions in KBEs.

The proposed project is conceived as the centerpiece for the PI’s work in the coming years. A large NSF ITR pre-proposal is currently pending that would support the collaboration efforts associated with this work. Another proposal to NSF’s CSS program would continue our OMOL project, orienting it particularly toward the deployment and adoption effort at the corporate study sites and at the analysis of the social impact of the KBE software in these settings, while a pending CISE Research Infrastructure proposal includes equipment and infrastructure for the local academic study sites. If funded, these other proposals would provide adequate support to fully achieve the vision proposed here. If they are not funded, the PI will seek other grants leveraged on the work proposed here.

The proposed work is designed to provide some of the basic conditions for the sorts of intensive collaboration that we believe are necessary to achieve the potential of KBE technology. In addition to providing technical conditions (e.g., Server, architecture, standards) and social conditions (local, national, international networks), the work will prepare specific examples of functionality (e.g., computational Perspectives, automated linking) and concrete analyses of application. These results will be disseminated through communication channels including specific related conferences.

Within the PI's home institutions the proposed work will be an important catalyst for building an interdisciplinary collaborative effort of undergraduates, graduate students, research staff, and faculty working on KBE theory, technology, and evaluation. The proposed work defines a coherent research agenda over several years that will crystallize a local research community.

References Cited

Bakhtin, M. (1986) *Speech Genres and Other Late Essays,* (V. McGee, Trans.), University of Texas Press, Austin, TX.

Boborow, D. & Goldstein, I. (1980) *An experimental description-based programming environment: Four reports,* Technical Report No. CSL-81-3, Xerox Palo Alto Research Center, Palo Alto, CA.

Boland, R. J. & Tenkasi, R. V. (1995) Perspective making and perspective taking in communities of knowing, *Organization Science,* 6 (4), pp. 350-372.

Bourdieu, P. (1972/1995) *Outline of a Theory of Practice,* (R. Nice, Trans.), Cambridge University Press, Cambridge, UK.

Brown, A. & Campione, J. (1994) Guided discovery in a community of learners. In K. McGilly (Ed.) *Classroom Lessons: Integrating Cognitive Theory and Classroom Practice,* MIT Press, Cambridge, MA, pp. 229-270.

Brown, J. S. & Duguid, P. (1991) Organizational learning and communities-of-practice: Toward a unified view of working, learning, and innovation, *Organization Science,* 2 (1), pp. 40-57.

Buckingham Shum, S. & Hammond, N. (1994) Argumentation-based design rationale: What use at what cost?, *International Journal of Human-Computer Studies,* 40 (4), pp. 603-652.

Chaiklin, S. & Lave, J. (Eds.) (1993) *Understanding Practice: Perspectives on Activity and Context,* Cambridge University Press, New York, NY.

Cole, M. (1996) *Cultural Psychology,* Harvard University Press, Cambridge, MA.

Connolly, T. (1997) Electronic brainstorming: Science meets technology in the group meeting room. In S. Kiesler (Ed.) *Culture of the Internet,* Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 263-276.

Cuthbert, A. (1999) Designs for collaborative learning environments: Can specialization encourage knowledge integration?, In: Proceedings of *Computer Supported Collaborative Learning (CSCL '99)*, Palo Alto, CA, pp. 117-126.

dePaula, R. (1998) *Computer Support for Collaborative Learning: Understanding Practices and Technology Adoption*, Masters Thesis, Telecommunications Department, University of Colorado, Boulder, CO.

Donald, M. (1991) Origins of the Modern Mind: Three Stages in the Evolution of Culture and Cognition, Harvard University Press, Cambridge, MA.

Donath, J., Karahalios, K., & Viegas, F. (1999) Visualizing conversation, *Journal of Computer Mediated Communication,* 4 (4). Available at: http://www.ascusc.org/jcmc/vol4/issue4/donath.html.

Engeström, Y., Miettinen, R., & Punamaki, R.-L. (Eds.) (1999) *Perspectives on Activity Theory,* Cambridge University Press, New York, NY.

Feltovich, P., Spiro, R., Coulson, R., & Feltovich, J. (1996) Collaboration within and among minds: Mastering complexity, individually and in groups. In T. Koschmann (Ed.) *CSCL: Theory and Practice of an Emerging Paradigm,* Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 25-44.

Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., & Sumner, T. (1993/1998) Embedding critics in design environments. Reprint from 1993; including Reflections from 1998. In M. T. Maybury & W. Wahlster (Eds.), *Readings in Intelligent User Interfaces,* Morgan Kaufman, New York, pp. 537-561. Available at: http://www.cs.colorado.edu/~gerry/publications/journals/ker/index.html.

Gadamer, H.-G. (1960/1988) *Truth and Method,* Crossroads, New York, NY.

Geertz, C. (1973) *The Interpretation of Cultures,* Basic Books, New York, NY.

Giddens, A. (1984) *The Constitution of Society,* U of California Press, Berkeley, CA.

Guzdial, M. & Turns, J. (2000) Sustaining discussion through a computer-mediated anchored discussion forum, *Journal of the Learning Sciences* .

Hatano, G. & Inagaki, K. (1991) Sharing cognition through collective comprehension activity. In L. Resnick, J. Levine, & S. Teasley (Eds.), *Perspectives on Socially-shared Cognition,* APA, Washington, DC, pp. 331-348.

Heidegger, M. (1927/1996) *Being and Time: A Translation of Sein und Zeit,* (J. Stambaugh, Trans.), SUNY Press, Albany, NY.

Hendricksen, C. (1999) Asynchronous collaboration by Document Review, *DocReview* . Available at: http://students.washington.edu/veritas/papers/DRpaper/basedoc.html.

Hewitt, J., Scardamalia, M., & Webb, J. (1998) *Situative design issues for interactive learning environments,* at http://csile.oise.on.ca/abstracts/situ_design.

Hewitt, J. & Teplovs, C. (1999) An analysis of growth patterns in computer conferencing threads, In: Proceedings of *Computer Supported Collaborative Learning (CSCL '99)*, Palo Alto, CA, pp. 232-241.

Hutchins, E. (1996) *Cognition in the Wold,* MIT Press, Cambridge, MA.

Jordan, B. & Henderson, A. (1995) Interaction analysis: Foundations and practice, *Journal of the Learning Sciences,* 4 (1), pp. 39-103.

Kintsch, E., Steinhart, D., Stahl, G., Matthews, C., Lamb, R., & the LSA Research Group (2000) Developing summarization skills through the use of LSA-backed feedback, *Interactive Learning Environments* .

Kling, R. (1999) What is social informatics and why does it matter?, *D-Lib Magazine,* 5 (1).

Kraemer, K. & Pinsonneault, A. (1990) Technology and groups: Assessment of the empirical research. In J. Galegher, R. Kraut, & C. Egido (Eds.), *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work,* Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 375-405.

Landauer, T. K. & 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), pp. 211-240.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998) Introduction to latent semantic analysis, *Discourse Processes,* 25 (1), pp. 259-284.

Lave, J. (1991) Situating learning in communities of practice. In L. Resnick, J. Levine, & S. Teasley (Eds.), *Perspectives on Socially Shared Cognition,* APA, Washington, DC, pp. 63-83.

Lave, J. & Wenger, E. (1991) *Situated Learning: Legitimate Peripheral Participation,* Cambridge University Press, Cambridge, UK.

McCall, R., Bennett, P., d'Oronzio, P., Ostwald, J., Shipman, F., & Wallace, N. (1990) PHIDIAS: Integrating CAD graphics into dynamic hypertext. In A. Rizk, N. Streitz, & J. Andre (Eds.), *Hypertext: Concepts, Systems and Applications. Proceedings of the European Conference on Hypertext (ECHT '90),* Cambridge University Press, Cambridge, UK, pp. 152-165.

McLean, R. (1999) Meta-communication widgets for knowledge building in distance education, In: Proceedings of *Computer Supported Collaborative Learning (CSCL '99)*, Palo Alto, CA, pp. 383-390.

Mittal, S., Bobrow, D., & Kahn, K. (1986) Virtual copies at the boundary between classes and instances, In: Proceedings of *Object-Oriented Programming Systems, Languages and Applications (OOPSLA '86)*, Portland, OR, pp. 159-166.

Muukkonen, H., Hakkarainen, K., & Lakkala, M. (1999) Collaborative technology for facilitating progressive inquiry: Future Learning Environment tools, In: Proceedings of *Computer Supported Collaborative Learning (CSCL '99)*, Palo Alto, CA, pp. 406-415.

Nardi, B. (Ed.) (1996) Context and Consciousness: Activity Theory and Human-Computer Interaction, MIT Press, Cambridge, MA.

Nelson, T. (1981) *Literary Machines,* Mindful Press, New York, NY.

Nelson, T. H. (1995) The heart of connection: Hypermedia unified by transclusion, *Communications of the ACM,* 38 (8), pp. 3-33.

Norman, D. A. (1993) *Things That Make Us Smart,* Addison-Wesley Publishing Company, Reading, MA.

Nygaard, K. & Sørgaard, P. (1987) The perspective concept in informatics. In G. Bjerknes, P. Ehn, & M. Kyng (Eds.), *Computers and Democracy: A Scandinavian Challenge,* Avebury, Aldershot, UK, pp. 371-393.

Papert, S. (1980) Mindstorms: Children, Computers and Powerful Ideas, Basic Books, New York, NY.

Pea, R. (1993) The collaborative visualization project, *Communications of the ACM,* 36 (5), pp. 60-63.

Peters, J. D. (1998) *Speaking Into the Air,* University of Chicago Press, Chicago, IL.

Roshelle, J. (1996) Learning by collaborating: Convergent conceptual change. In T. Koschmann (Ed.) *CSCL: Theory and Practice of an Emerging Paradigm,* Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 209-248.

Scardamalia, M. & 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 , pp. 37-68.

Scardamalia, M. & Bereiter, C. (1996) Computer support for knowledge-building communities. In T. Koschmann (Ed.) *CSCL: Theory and Practice of an Emerging Paradigm,* Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 249-268.

Slotta, J. D. & Linn, M. C. (2000) The Knowledge Integration Environment: Helping students use the Internet effectively. In M. J. Jacobson & R. Kozma (Eds.), *Learning the Sciences of the 21st Century,* Lawrence Erlbaum Associates, Hillsdale, NJ.

Stahl, G. (1993a) *Interpretation in Design: The Problem of Tacit and Explicit Understanding in Computer Support of Cooperative Design*, Ph.D. Dissertation, Department of Computer Science, University of Colorado, Boulder, CO. Available at: http://www.cs.colorado.edu/~gerry/publications/dissertations/dis_intro.html.

Stahl, G. (1993b) Supporting situated interpretation, In: Proceedings of *Annual Meeting of the Cognitive Science Society (CogSci '93)*, Boulder, CO, pp. 965-970. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1990-1997/cogsci93/CogSci.html.

Stahl, G. (1995) *Supporting Personalizable Learning,* Technical Report No. CU-CS-788-95, Department of Computer Science, University of Colorado, Boulder, CO. Available at: http://www.cs.colorado.edu/~gerry/publications/techreports/personalize/.

Stahl, G. (1997) poster: WebGuide, In: Proceedings of *Computer Support for Collaborative Learning (CSCL '97)*, Toronto, Canada.

Stahl, G. (1998a) Collaborative information environments for innovative communities of practice, In: Proceedings of *German Computer-Supported Cooperative Work conference (DCSCW '98)*, Dortmund, Germany. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1998/dcscw98/dcscw.html.

Stahl, G. (1998b) Presentation of WebGuide, In: Proceedings of *Annual conference of the Center for Innovative Learning Technologies (CILT '98)*, San Jose, CA.

Stahl, G. (1999a) demo: WebGuide: Computational perspectives for learning communities, In: Proceedings of *Annual conference of the Center for Innovative Learning Technologies (CILT '99)*, San Jose, CA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1999/cilt99/index.html.

Stahl, G. (1999b) POW! Perspectives on the Web, In: Proceedings of *WebNet World Conference on the WWW and Internet (WebNet '99)*, Honolulu, HA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1999/webnet99/webnet99.html.

Stahl, G. (1999c) Reflections on WebGuide: Seven issues for the next generation of collaborative knowledge-building environments, In: Proceedings of *Computer Supported Collaborative Learning (CSCL '99)*, Palo Alto, CA, pp. 600-610. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1999/cscl99/.

Stahl, G. (1999d) WebGuide: Guiding collaborative learning on the Web with perspectives, In: Proceedings of *Annual Conference of the American Educational Research Association (AERA '99)*, Montreal, Canada. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1999/aera99/.

Stahl, G. (2000a) book review: Professional Development for Cooperative Learning: Issues and Approaches, *Teaching and Learning in Medicine: An International Journal* . Available at: http://www.cs.colorado.edu/~gerry/publications/journals/medicine/coop_learn.html.

Stahl, G. (2000b) Collaborative information environments to support knowledge construction by communities, *AI & Society,* 14 , pp. 1-27. Available at: http://www.cs.colorado.edu/~gerry/publications/journals/ai&society/.

Stahl, G. (2000c) A model of collaborative knowledge-building, In: Proceedings of *International Conference of the Learning Sciences (ICLS 2000)*, Ann Arbor, MI. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/2000/icls/.

Stahl, G. & dePaula, R. (1998) poster: Learning perspectives, In: Proceedings of *International Conference on the Learning Sciences (ICLS'98)*, Atlanta, GA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1998/icls98/icls98.html.

Stahl, G., dePaula, R., & the LSA Group (2000) Evolution of an LSA-based interactive environment for learning to write summaries, *Interactive Learning Environments* . Available at: http://www.cs.colorado.edu/~gerry/publications/journals/ILE/ile.html.

Stahl, G., Fischer, G., Nakakoji, K., Ostwald, J., & Sumner, T. (1993) Embedding computer-based critics in the contexts of design, In: Proceedings of *Conference on Human Factors in Computing Systems (INTERChi '93)*, Amsterdam, Holland, pp. 157-164. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1990-1997/chi93/CHI93.html.

Stahl, G., Hermann, T., dePaula, R., & Loser, K.-U. (1998a) demo: WebGuide: Guiding cooperative work on the Web with support for perspectives and negotiation, In: Proceedings of *Computer Supported Cooperative Work (CSCW '98)*, Seattle, WA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1998/cscw98/cscw_demo.html.

Stahl, G. & Herrmann, T. (1998) Verschrankung von Perspectiven durch Aushandlung (in German; translated by G. Stahl as: The sharing of perspectives by means of negotiation), In: Proceedings of *Interaktion in Web: Innovative Kommunikationsformen*, Marburg, Germany. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1998/verschrankung/index.html and http://www.cs.colorado.edu/~gerry/publications/conferences/1998/sharing/sharing.html.

Stahl, G. & Herrmann, T. (1999) Intertwining perspectives and negotiation, In: Proceedings of *International Conference on Supporting Group Work (Group '99)*, Phoenix, AZ. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1999/group99/.

Stahl, G. & Hoadley, C. (2000) workshop: Strategies for sustaining interaction in online discussion forums and virtual communities, In: Proceedings of *Annual Conference of the American Educational Research Association (AERA 2000)*, New Orleans, LA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/2000/aera2000/aera.html.

Stahl, G. & Koschmann, T. (1998) panel: Learning issues in problem-based learning: Situating collaborative information seeking, In: Proceedings of *Computer Supported Cooperative Work (CSCW '98)*, Seattle, WA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1998/cscw98/cscw_workshop.html.

Stahl, G., Koschmann, T., & Ostwald, J. (1998b) workshop: Shouldn't we really be studying practice?, In: Proceedings of *International Conference on the Learning Sciences (ICLS '98)*, Atlanta, GA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1998/icls98/ICLS Workshop.html.

Stahl, G., Koschmann, T., & Scardamalia, M. (1999) workshop: Collaborating on the design and assessment of knowledge-building environments in the 2000's, In: Proceedings of *Computer Support for Collaborative Learning (CSCL '99)*, Stanford, CA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1999/cscl99/cscl99_workshop.html.

Stahl, G., Sumner, T., & 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), pp. 237-246. Available at: http://www.cs.colorado.edu/~gerry/publications/journals/c&e/.

Steinhart, D. (2000) *Designing, Testing, and Evaluating Summary Street, an LSA-Based Summarization Tool (in preparation)*, Ph.D. Dissertation, Department of Psychology, University of Colorado, Boulder, CO.

Sumner, T. & Buckingham Shum, S. (1998a) From documents to discourse: Shifting conceptions of scholarly publishing, In: Proceedings of *Computer-Human Interaction (CHI 98)*, Los Angeles, CA, pp. 95-102.

Sumner, T. & Buckingham Shum, S. (1998b) A toolkit for publishing Web discussion documents: Design principles and case studies, In: Proceedings of *Asia Pacific Computer Human Interaction (APCHI 98)*, Shonan Village Center, Japan, pp. 218-223.

Suthers, D. (1999) Representational support for collaborative inquiry, In: Proceedings of *32nd Hawai'i International Conference on the System Sciences (HICSS-32),*, Maui, HA, pp. CD-ROM. Available at: http://lilt.ics.hawaii.edu/lilt/papers/hicss99.pdf.

Vogel, D., Nunamaker, J., Applegate, L., & Konsynski, B. (1987) Group decision support systems: Determinants of success, In: Proceedings of *Decision Support Systems (DSS '87)*, pp. 118-128.

Vygotsky, L. (1930/1978) *Mind in Society,* Harvard University Press, Cambridge, MA.

Wenger, E. (1987) *Artificial Intelligence and Tutoring Systems,* Morgan Kaufman, Los Altos, CA.

Wenger, E. (1998) *Communities of Practice,* Cambridge University Press, Boston, MA.

Winograd, T. & Flores, F. (1986) Understanding Computers and Cognition: A New Foundation of Design, Addison-Wesley, Reading, MA.

RESUME

Center for LifeLong Learning and Design
Department of Computer Science, and
Institute of Cognitive Science
University of Colorado, Boulder, CO 80309-0430

(303) 492-3912 (phone)
(303) 492-2844 (fax)
Gerry.Stahl@Colorado.edu
www.cs.colorado.edu/~gerry

EDUCATION

	University of Colorado
1996-99	Postdoctoral Research Fellow
1993	Ph.D. in Computer Science
1990	M.Sc. in Computer Science
	Northwestern University
1975	Ph.D. in Philosophy
1971	M.A. in Philosophy
	University of Frankfurt
1973	Graduate study in critical social theory
	University of Heidelberg
1968	Graduate study in continental philosophy
	Massachusetts Institute of Technology (MIT)
1967	B.S. in Humanities & Science (Math & Philosophy)

PROFESSIONAL EXPERIENCE

	Research Professor
1999-present	Department of Computer Science and Institute of Cognitive Science, Boulder, CO
	Post Doctoral Research Fellow
1996-1999	Center for LifeLong Learning and Design, Boulder, CO
	President
1995-1996	Personalizable Software, Niwot, CO
	Director of Software R&D
1993-1996	Owen Research Inc., Boulder, CO
	Graduate Research Assistant
1990-1993	College of Environmental Design, Boulder, CO
	Intern Interface Developer
1990-1991	US West Advanced Technology, Denver & Boulder, CO
	Computer Science Instructor & Teaching Assistant
1989-1990	University of Colorado, Boulder, CO
	Executive Director
1984-1989	Community Computerization Project, Philadelphia, PA
	Planning and Evaluation Specialist
1979-1984	Southwest Germantown Community Devel. Corp., Philadelphia, PA
	Community Organizer & VISTA Supervisor
1978-1979	Philadelphia Council of Neighborhood Organizations, Philadelphia, PA
	Systems Programmer
1974-1977	Temple University, Philadelphia, PA
1970-1971	Northwestern University, Evanston, IL
1969-1970	Temple University, Philadelphia, PA
	Applications Programmer
Summer 1966	Brown Bovari Cie, Baden, Switzerland
Summer 1965	University of Pennsylvania, Philadelphia, PA

Current Grants

- 1997-2000: "Allowing Learners to be Articulate: Incorporating Automated Text Evaluation into Collaborative Software Environments" (primary author and primary software developer; PIs: Gerhard Fischer, Walter Kintsch and Thomas Landauer) $678,239; Sponsor: James S. McDonnell Foundation, Cognitive Science in Education Program.
- 1997-2000: "Conceptual Frameworks and Computational Support for Organizational Memories and Organizational Learning" (co-PI with Gerhard Fischer and Jonathan Ostwald), $725,000; Sponsor: NSF, Computation and Social Systems program.
- 1999-2000: "Interoperability among Knowledge Building Environments" (PI) $9,124; Sponsor: Center for Innovative Learning Technology / SRI.
- 1998-1999: "Collaborative Web-Based Tools for Learning to Integrate Scientific Results into Social Policy" (co-PI with Ray Habermann) $89,338; Sponsor: NSF.

Pending Grant Proposals

- Intel Corporation: $191,100 over three years to develop computational perspectives mechanisms.
- Colorado Advanced Software Institute: $40,000 over one year to develop a perspectives server.
- Lotus Corporation: $50,000 over one year to study models of collaboration.
- NSF ITR program: $2,700,000 over five years to foster national and international collaboration on knowledge-building environments.
- NSF ITR program: $500,000 over three years to study the use of computational perspectives in collaborative knowledge-building environments.
- NSF CSS program: $400,000 over three years to study the use of collaborative knowledge-building environments in various field sites.
- NSF ROLE program: $600,000 over three years to study the activities involved in collaboration using conversational analysis methods.

Software Development Projects (1990-1999)

These are some of the systems I implemented in the past decade, either at the University of Colorado or as a consultant. Except where indicated, I was the primary developer and they resulted in working prototypes. Theoretical frameworks for most of them are presented in my *Selected Writings*. In some cases (e.g., `State the Essence` and `WebGuide`) they are being used in on-going field studies.

`Hermes`: a hypermedia substrate for design environments, including computational perspectives, persistent versions of drawings, user-programmable critic agents, extensible scripting language. This was my dissertation system. I used it to implement an environment for lunar habitat design. NASA subsequently used it for their *Space Station Freedom Man-Systems Integration Standards* design guidelines for spacecraft.

`Crew`: a case-based reasoning system to model long-term astronaut missions, developed under contract with NASA in cooperation with psychologists at Houston astronaut support. Extends and integrates several AI techniques.

`TCA`: a digital library for teachers to exchange, discuss, and adapt curricular materials for constructivist classrooms. Anticipated approaches now being implemented years later. I prototyped this under an NSA SBIR grant.

`OptoNet`: a software feedback system in LabView with real-time analog-to-digital data acquisition to stabilize holographic equipment during space flight. My software was tested on NASA zero-g simulation flights.

`InterView`: an information management system for US West service providers to access data in multiple legacy database systems. I worked on C++ interface components to an object-oriented database management system.

`Tracker`: an information management system for the Baltimore public schools to track services to students on welfare. I worked on interface components to an Access database.

`CIE`: a workplace documentation support system for ISO 9000 certification, using computational perspectives to support bottom-up documentation processes.

`WebNet`: a Web-based design environment for LAN design and management. Integrated a number of components for collecting, displaying, up-dating and discussing domain knowledge and local knowledge about LAN configuration.

`State the Essence`: educational software using latent semantic analysis (LSA) to provide feedback to students summarizing a text. Used in middle school classrooms for three years and subjected to rigorous controlled experiments showing positive results, particularly for students having trouble understanding the given text.

`WebGuide`: a threaded discussion medium with personal and group perspectives to facilitate collaboration and management of shared information spaces. Tried out in a middle school course and a graduate seminar. Provides a software foundation for my future research.

TEACHING AND RESEARCH INTERESTS

Teaching and research are intimately related for me because:

- I believe that teaching can be most effective when it is situated within a context of authentic research issues and experiences.
- The most valuable thing a teacher can offer in a research university is his personal perspective on a field and contact with his research activities.
- My research is about learning theory and educational technology; and conversely
- My teaching is about the theory and design of educational technology, and my instructional approach incorporates this technology and experiments with it.

I treasure teaching: designing a coherent course, preparing content resources, establishing contact with students, sharing perspectives, and collaborating in processes of discovery and learning. Last year I voluntarily taught two semesters of a graduate seminar I created entitled "Readings and Research in Cognitive Science." Although as a Research Professor I am not required to teach and am not reimbursed for it, I offered this course as a venue for developing, testing, and sharing my research ideas. (The course subsequently became a core course in the new interdisciplinary cognitive science doctoral program.) The first semester was a readings course on learning theory, starting with Lave's social formulation of situated learning and tracing its core concept of "mediation" back through cultural psychology to the philosophy of Hegel and Marx. Class discussions of the readings and course themes were mediated by my `WebGuide` software. The second semester was a project-based course on "knowledge-building environments," involving education and computer science students in Colorado and in Germany, supported by a variety of Web-based technologies. The theoretical investigations and software development projects pursued in these seminars and the experience gained with the use of `WebGuide` set the agenda for my future work.

A theory of creativity and learning that I have been trying to formulate grows out of the approach of Piaget and Vygotsky: High-level cognitive abilities are not biologically predefined, but develop through personal developmental and social historical processes. I hypothesize that Web-based technology can unleash new forms of social cognition such as collaborative creativity and organizational learning. However, my experience to date indicates that the design of appropriate enabling technologies is a complex, wicked problem requiring research in collaboration, software design, and social informatics (e.g., deployment, acceptance, social practices).

My research approach is to analyze the activities involved in collaborative knowledge-building through microanalysis (conversational analysis) of knowledge-constituting discourses – both face-to-face and computer-mediated. Then I try to develop software functionality to support the various identified activities. For several years I have been experimenting with systems to support personal and group perspectives, which are critical to comprehend how multiple "voices" interact to produce a synergistic shared understanding. Now I want to integrate support for negotiation processes into knowledge-building environments with computational perspectives (like `WebGuide`).

Future research will deploy new versions of knowledge-building environments in diverse test sites: research groups, classrooms, industrial design teams, corporate training workshops, etc. Analysis of these experiences will feed back into the theoretical framework and the software design. The entire research agenda – from theory building through shrink-wrapped deployment – exceeds the scope of any one research group: The `CSILE` project and their colleagues, for instance, have already put an impressive effort into this for over a decade. Therefore, I have begun to organize a national network of researchers interested in knowledge-building environments, with the aim of collaborating with similar networks already existing in a number of other countries. I foresee collaborations interacting at various levels, from my students and co-workers to international networks, all supported by the knowledge-building environments we are constructing.

I have tried to summarize concisely here the core of my teaching and research agenda as it now stands. More detailed discussions are contained in my *Selected Writings*. Naturally, my intellectual and teaching interests are much broader than this core focus. I participate in courses and interact with students in computer science, cognitive science, and Web-based educational technology more generally. I remain eager to incorporate in my teaching and research new theoretical insights from innovative intellectual paradigms and to pursue potential practical applications to meet pressing social needs.

COMMUNITY DEVELOPMENT EXPERIENCE

Executive Director

1985-1988 Community Computerization Project, Philadelphia, PA

Provided computer services to 36 community nonprofit organizations, including development of customized client tracking and fund accounting systems. Conducted needs assessment for initial computerization, installed equipment, trained staff, customized software.

Computer Specialist

1984-1986 Institute for the Study of Civic Values, Philadelphia, PA

Initiated a computerization service to non-profit and community-based organizations during the period when personal computers were just becoming available.

Planning and Evaluation Specialist

1979-1984 Southwest Germantown Community Development Corporation, Philadelphia, PA

Neighborhood planner and development manager. Increased and diversified funding base from one source, $150,000 per year, to about $1,000,000 per year from federal, state, city, foundations, corporations, and generated income. Awarded several prestigious national grants from CBS, HUD, CETA for economic development, housing, youth employment projects. Also wrote grant proposals funded at over $2,000,000 for other community organizing, neighborhood improvement, energy conservation, community credit union, and technical assistance organizations. Duties included serving as assistant director in charge of grant management. Evaluated programs and submitted all funding-source reports, both programmatic and fiscal. Handled financial planning and managed multiple funds for organization and its subsidiaries. Also served as volunteer treasurer or paid bookkeeper for several other non-profit organizations.

Community Organizer & VISTA Supervisor

1978-1979 Philadelphia Council of Neighborhood Organizations, Philadelphia, PA

Established grassroots campaigns and organizations to address social needs in poor neighborhoods. Provided technical assistance to neighborhood groups and coordinate efforts at a citywide level. Trained and supervised community organizers in inner-city neighborhoods. Developed proposals for federal funds for neighborhood-based enterprises in energy conservation, weatherization and recycling (over $1,000,000 per year).

Research Associate

1977-1978 Philadelphia Unemployment Project, Philadelphia, PA

Researched causes of urban unemployment and planned advocacy and service programs for the unemployed.

Writings

Dissertations

Stahl, G. (1975) *Marxian Hermeneutics and Heideggerian Social Theory: Interpreting and Transforming Our World*, Ph.D. Dissertation, Department of Philosophy, Northwestern University, Evanston, IL. Available at: http://www.cs.colorado.edu/~gerry/publications/dissertations/thesis.htm.

Stahl, G. (1993) *Interpretation in Design: The Problem of Tacit and Explicit Understanding in Computer Support of Cooperative Design*, Ph.D. Dissertation, Department of Computer Science, University of Colorado, Boulder, CO. Available at: http://www.cs.colorado.edu/~gerry/publications/dissertations/dis_intro.html.

Journal Publications

Stahl, G. (1975) The jargon of authenticity: An introduction to a Marxist critique of Heidegger, *Boundary 2,* III (2), pp. 489-498. Available at: http://www.cs.colorado.edu/~gerry/publications/interpretations/jargon.htm.

Stahl, G. (1976) Attuned to Being: Heideggerian music in technological society, *Boundary 2,* IV (2), pp. 637-664. Available at: http://www.cs.colorado.edu/~gerry/publications/interpretations/attuned.htm.

Stahl, G., Fischer, G., Nakakoji, K., Ostwald, J., & Sumner, T. (1993/1998) Embedding critics in design environments. Reprint from 1993 (*Knowledge Engineering Review* (4) 8, 285-307); including Reflections from 1998. In M. T. Maybury & W. Wahlster (Eds.), *Readings in Intelligent User Interfaces,* Morgan Kaufman, New York, pp. 537-561. Available at: http://www.cs.colorado.edu/~gerry/publications/journals/ker/index.html.

Stahl, G., Sumner, T., & 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), pp. 237-246. Available at: http://www.cs.colorado.edu/~gerry/publications/journals/c&e/.

Stahl, G. (1996) Armchair missions to Mars: Using case-based reasoning and fuzzy logic to simulate a time series model of astronaut crews. (Forthcoming in Sankar Pal, Daniel So, Tharam Dillon (Eds.) (2000) *Soft Computing in Case Based Reasoning*, Springer Verlag), *Knowledge-Based Systems,* 9 , pp. 409-415. Available at: http://www.cs.colorado.edu/~gerry/publications/journals/crew/index.html.

Stahl, G. (2000) book review: Professional Development for Cooperative Learning: Issues and Approaches, *Teaching and Learning in Medicine: An International Journal* . Available at: http://www.cs.colorado.edu/~gerry/publications/journals/medicine/coop_learn.html.

Stahl, G. (2000) Collaborative information environments to support knowledge construction by communities, *AI & Society,* 14 , pp. 1-27. Available at: http://www.cs.colorado.edu/~gerry/publications/journals/ai&society/.

Stahl, G. & dePaula, R. (2000) Evolution of an interactive medium for learning to write summaries (in preparation), *Journal of Interactive Media in Education (JIME)* . Available at: http://www.cs.colorado.edu/~gerry/publications/journals/ILE/ile.html.

Conference papers

Stahl, G., McCall, R., & Peper, G. (1992) Extending hypermedia with an inference language: An alternative to rule-based expert systems, In: Proceedings of *IBM ITL Conference: Expert Systems*, pp. 160-167. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1990-1997/ibm92/ExtHyper.html.

Stahl, G. (1993) Supporting situated interpretation, In: Proceedings of *Annual Meeting of the Cognitive Science Society (CogSci '93)*, Boulder, CO, pp. 965-970. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1990-1997/cogsci93/CogSci.html.

Stahl, G., Fischer, G., Nakakoji, K., Ostwald, J., & Sumner, T. (1993) Embedding computer-based critics in the contexts of design, In: Proceedings of *Conference on Human Factors in Computing Systems (INTERChi '93)*, Amsterdam, Holland, pp. 157-164. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1990-1997/chi93/CHI93.html.

Stahl, G., Sumner, T., & Repenning, A. (1995) Internet repositories for collaborative learning: Supporting both students and teachers, In: Proceedings of *Computer Support for Collaborative Learning (CSCL '95)*, Bloomington, Indiana, pp. 321-328. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1990-1997/cscl95/cscl.htm

Stahl, G. (1998) Collaborative information environments for innovative communities of practice, In: Proceedings of *German Computer-Supported Cooperative Work Conference (D-CSCW '98): Groupware und organizatorische*

Innovation, Dortmund, Germany, pp. 195-210. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1998/dcscw98/dcscw.html.

Stahl, G. & Herrmann, T. (1998) Verschrankung von Perspectiven durch Aushandlung (in German; translated by G. Stahl as: The sharing of perspectives by means of negotiation), In: Proceedings of *Interaktion in Web: Innovative Kommunikationsformen*, Marburg, Germany, pp. 95-112. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1998/verschrankung/index.html and http://www.cs.colorado.edu/~gerry/publications/conferences/1998/sharing/sharing.html.

Stahl, G. (1999) WebGuide: Guiding collaborative learning on the Web with perspectives, In: Proceedings of *Annual Conference of the American Educational Research Association (AERA '99)*, Montreal, Canada. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1999/aera99/.

Stahl, G. (1999) POW! Perspectives on the Web, In: Proceedings of *WebNet World Conference on the WWW and Internet (WebNet '99)*, Honolulu, HA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1999/webnet99/webnet99.html.

Stahl, G. & Herrmann, T. (1999) Intertwining perspectives and negotiation, In: Proceedings of *International Conference on Supporting Group Work (Group '99)*, Phoenix, AZ. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1999/group99/.

Stahl, G. (1999) Reflections on WebGuide: Seven issues for the next generation of collaborative knowledge-building environments, In: Proceedings of *Computer Supported Collaborative Learning (CSCL '99)*, Palo Alto, CA, pp. 600-610. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1999/cscl99/.

Stahl, G. (2000) A model of collaborative knowledge-building, In: Proceedings of *International Conference of the Learning Sciences (ICLS 2000)*, Ann Arbor, MI. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/2000/icls/.

Conference Presentations

Stahl, G. (1997) poster: WebGuide, In: Proceedings of *Computer Support for Collaborative Learning (CSCL '97)*, Toronto, Canada.

Stahl, G. (1998) Presentation of WebGuide, In: Proceedings of *Annual conference of the Center for Innovative Learning Technologies (CILT '98)*, San Jose, CA.

Stahl, G. & dePaula, R. (1998) poster: Learning perspectives, In: Proceedings of *International Conference on the Learning Sciences (ICLS'98)*, Atlanta, GA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1998/icls98/icls98.html.

Stahl, G., Hermann, T., dePaula, R., & Loser, K.-U. (1998) demo: WebGuide: Guiding cooperative work on the Web with support for perspectives and negotiation, In: Proceedings of *Computer Supported Cooperative Work (CSCW '98)*, Seattle, WA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1998/cscw98/cscw_demo.html.

Stahl, G. & Koschmann, T. (1998) panel: Learning issues in problem-based learning: Situating collaborative information seeking, In: Proceedings of *Computer Supported Cooperative Work (CSCW '98)*, Seattle, WA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1998/cscw98/cscw_workshop.html.

Stahl, G., Koschmann, T., & Ostwald, J. (1998) workshop: Shouldn't we really be studying practice?, In: Proceedings of *International Conference on the Learning Sciences (ICLS '98)*, Atlanta, GA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1998/icls98/ICLS Workshop.html.

Stahl, G., Koschmann, T., & Scardamalia, M. (1999) workshop: Collaborating on the design and assessment of knowledge-building environments in the 2000's, In: Proceedings of *Computer Support for Collaborative Learning (CSCL '99)*, Stanford, CA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1999/cscl99/cscl99_workshop.html.

Stahl, G. (1999) demo: WebGuide: Computational perspectives for learning communities, In: Proceedings of *Annual conference of the Center for Innovative Learning Technologies (CILT '99)*, San Jose, CA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1999/cilt99/index.html.

Stahl, G. & Hoadley, C. (2000) workshop: Strategies for sustaining interaction in online discussion forums and virtual communities, In: Proceedings of *Annual Conference of the American Educational Research Association (AERA 2000)*, New Orleans, LA. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/2000/aera2000/aera.html.

Technical Reports

Stahl, G. (1991) *A Hypermedia Inference Language as an Alternative to Rule-based Systems,* Technical Report No. CU-CS-557-91, Department of Computer Science, University of Colorado, Boulder, CO. Available at: http://www.cs.colorado.edu/~gerry/publications/conferences/1990-1997/ibm92/InfLang.html.

Stahl, G. (1992) *A Computational Medium for Supporting Interpretation in Design,* Technical Report No. CU-CS-598-92, Department of Computer Science, University of Colorado, Boulder, CO. Available at: http://www.cs.colorado.edu/~gerry/publications/techreports/design/Design.tr.html.

Stahl, G. (1992) *Toward a Theory of Hermeneutic Software Design: Dissertation Proposal,* Technical Report No. CU-CS-589-92, Department of Computer Science, University of Colorado, Boulder, CO. Available at: http://www.cs.colorado.edu/~gerry/publications/dissertations/Proposal.html.

Stahl, G. (1993) *Computer Support for Situated, Perspectival, Linguistic Interpretation in Non-Routine Design*, Abridged version of dissertation, Department of Computer Science, University of Colorado, Boulder, CO.

Stahl, G. (1995) *Supporting Personalizable Learning,* Technical Report No. CU-CS-788-95, Department of Computer Science, University of Colorado, Boulder, CO. Available at: http://www.cs.colorado.edu/~gerry/publications/techreports/personalize/.

Stahl, G. (1996) *Personalizing the Web,* Technical Report No. CU-CS-836-96, Department of Computer Science, University of Colorado, Boulder, CO. Available at: http://www.cs.colorado.edu/~gerry/publications/techreports/www6/PAPER82.html.

Stahl, G. (1996) *Supporting Interpretation in Design,* Technical Report No. CU-CS-598-92, Department of Computer Science, University of Colorado, Boulder, CO. Available at: http://www.cs.colorado.edu/~gerry/publications/techreports/design/Design.tr.html.