University of Colorado at Boulder

Department of Computer Science

ECOT 7-7 Engineering Center
Campus Box 430
Boulder, Colorado 80309-0430
(303) 492-7514, FAX: (303) 492-2844

Invited Talk, EW-ED'94, East-West Conference on Computer Technologiesin Education, September 1994, Crimea, Ukraine

Computational Environments in Support of Self-Directed Learning

Gerhard Fischer

Department of Computer Science and Institute of Cognitive Science
Campus Box 430, University of Colorado, Boulder, CO 80309
gerhard@cs.colorado.edu

ABSTRACT

Information overload, the advent of high-functionality systems, and a climate of rapid technological change have created new problems and challenges for education and training. New instructional approaches are needed to circumvent the difficult problems of *coverage* (i.e., trying to teach people everything that they may need to know in the future) and *obsolescence* (i.e., trying to predict what specific knowledge someone will need in the future). *Self-directed learning* addresses these problems in the following way: (1) it contextualizes learning by allowing it to be integrated into work rather than relegating it to a separate phase; (2) it lets learners see for themselves the usefulness of new knowledge for actual problem situations, thereby increasing the motivation for learning new skills and information; and (3) it makes new information relevant to the task at hand.

Creating computational environments in support of self-directed learning generates a large number of challenging problems. We have developed *domain-oriented design environments* in support of self-directed learning. They represent learner-centered alternatives to teacher-centered tutoring systems and they augment open-ended, unsupported learning environments by providing advice, assistance, and guidance if needed in breakdown situations.

Acknowledgments

The author would like to thank the members of the Human-Computer Communication group at the University of Colorado who contributed substantially to the conceptual framework and the systems discussed in this paper. The research was supported the (1) the National Science Foundation under grant MDR-9253245, (2) NYNEX Science and Technology Center (White Plains, NY), and (3) Software Research Associates, Inc. (Tokyo, Japan).

Computational Environments in Support of Self-Directed Learning

Gerhard Fischer
Department of Computer Science and Institute of Cognitive Science
University of Colorado, Campus Box 430
Boulder, Colorado 80309 USA
gerhard@cs.colorado.edu

Abstract. Information overload, the advent of high-functionality systems, and a climate of rapid technological change have created new problems and challenges for education and training. New instructional approaches are needed to circumvent the difficult problems of *coverage* (i.e., trying to teach people everything that they may need to know in the future) and *obsolescence* (i.e., trying to predict what specific knowledge someone will need in the future). *Self-directed learning* addresses these problems in the following way: (1) it contextualizes learning by allowing it to be integrated into work rather than relegating it to a separate phase; (2) it lets learners see for themselves the usefulness of new knowledge for actual problem situations, thereby increasing the motivation for learning new skills and information; and (3) it makes new information relevant to the task at hand.

Creating computational environments in support of self-directed learning generates a large number of challenging problems. We have developed *domain-oriented design environments* in support of self-directed learning. They represent learner-centered alternatives to teacher-centered tutoring systems and they augment open-ended, unsupported learning environments by providing advice, assistance, and guidance if needed in breakdown situations.

1 Introduction

Innovative uses of computers in education have focused on two major approaches: intelligent tutoring systems [Polson, Richardson 88; Wenger 87] and open learning environments [Papert 80; Boecker, Eden, Fischer 91].

The strength of *intelligent tutoring systems* (ITSs) lies in their ability to teach basic concepts and skills of a problem domain. However, the problems presented by tutoring systems are prespecified by the authors of the systems rather than being ill-defined and arising out of real-world contingencies. Thus, tutoring systems, similar to school education, leave it to the learner to relate training to real-world problem situations. *Interactive learning environments* (ILEs) avoid the problem of presenting instructional material in a system-controlled order without regard to the learner's situation, but they only provide limited support in helping learners detect mistakes or overcome breakdowns [Fischer 94]. Misconceptions may accumulate into chains, in which each later misconception is based on a previous one. Learners get trapped on suboptimal plateaus because they fail to discover the knowledge needed to make better use of their tools and to create better artifacts.

For many years we have been trying to find new ways of creating computer-based learning environments, combining the strengths of both approaches while at the same time eliminating some of the weaknesses (see Figure 1). Our goal has been to support *self-directed learning* by (1) providing opportunities for helping people to take charge of their own learning, (2) letting learners develop their own designs, representations, models and arguments, and (3) conceptualizing learning as a lifelong process driven by being alive and in the world.

Domain-oriented design environments (DODEs) [Fischer et al. 91]) have proven to be powerful and versatile environments for learning that (1) address the limitations of intelligent tutoring systems and interactive learning environments, and (2) provide multiple learning opportunities (see Figure 1). Pursuing this line of research, we have emphasized AI research directions and techniques that augment and complement human intelligence with rich computational environments, including critics, agents, assistants, adaptable and adaptive tools, information access and information delivery mechanisms [Nakakoji, Fischer 94].

2 Problems Addressed by Our Work

Deschooling Learning. Learning should be part of living, a natural consequence of being alive and in touch with the world, and not a process separate from the rest of life [Resnick 89; Lave 88]. What learners need, therefore, is not only instruction but *access* to the world (in order to connect the knowledge in their head with the knowledge in the world [Norman 93]) and a chance to play a meaningful part in it. Education should be a

	ITSs	DoDEs	ILEs
support for understanding basic concepts and skills	high	moderate	absent
degree to which activity directed by learner/user	low	high	high
scaffolding and support for breakdowns	moderate	moderate	absent
domain knowledge	deep for specific problems	for many problems within a domain	none

Figure 1: A Comparison between ITSs, DoDEs and ILEs

distributed lifelong process by which one learns material as one needs it. School learning and work place learning need to be integrated.

We refer to work place learning not as it is currently practiced (e.g., companies imitating school learning by sending their employees to decontextualized classrooms), but as it could be or should be. Examples include apprenticeship type relationships such as internships for doctors and PhD studies. In such learning situations, problems are not given, but need to be framed. Collaboration is critical and learning is firmly integrated with working. Figure 2 compares school and work place learning along a number of dimensions.

	Schools	**Workplace**
EMPHASIS ON:	"basic" skills	education embedded in ongoing work activities
POTENTIAL DRAWBACKS:	decontextualized, not situated	important concepts are not encountered
PROBLEMS:	given	constructed
NEW TOPICS: DEFINED BY:	curricula	arise accidentally from work situations
STRUCTURE:	pedagogic or "logical" structure	work activity
ROLES:	expert ↔ novice model	reciprocal learning
TEACHERS/ TRAINERS:	expound subject matter	engage in work practice
MODE:	instructionism (knowledge absorption)	constructionism (knowledge construction)

Figure 2: Integration of School and Work Place Learning

Information Overload. Information overload (further increased by world-wide webs of electronic information), the advent of high-functionality systems, and a climate of rapid technological change have created new problems and challenges for education and training. New information technologies must take into account not only the producers of that information but also its consumers. Providing more information will not make computers more helpful; instead, there is a need for systems that help us attend to the most useful, most interesting, or most valuable information [Fischer 91]. New instructional approaches are called for to solve the difficult problems of *coverage* (trying to teach people everything that they may need to know in the future) and *obsolescence* (trying to predict what specific knowledge someone will need in the future). Most traditional approaches to learning are based on a model in which one attends school or college to prepare for the situations that will occur in later life. These approaches are doomed to fail because there is too much to learn and because the content that is learned becomes outdated too quickly.

Beyond Passive Tools. Passive tools do not support mixed-initiative dialogs, such as dialogs in which either the learner or the system may initiate an interaction or a change of topic. Passive tools are often inadequate for the demands of situations in which we envision our systems being used. For example, browsing does not scale up to large information spaces, and passive help systems are of little use if learners do not know that relevant information exists. Information access mechanisms must be combined with information delivery mechanisms [Nakakoji, Fischer 94]. Passive tools must be complemented by intelligent agents, such as critics [Fischer et al. 91]), which act on their own initiative.

Production Paradox. Educational approaches must avoid the "production paradox" [Carroll, Rosson 87], in which learning is inhibited by lack of time and working is inhibited by lack of knowledge. Learners must regard the time and effort invested in learning as immediately worthwhile for the task at hand—not as valuable merely for some putative long-term gain.

Insufficient Attention to Motivation. Many approaches to learning have paid little attention to motivation. Ignoring motivational issues [Csikszentmihalyi 90; Norman 93] will make even the most technologically sophisticated efforts fail. To motivate learners, we advocate the following: (1) learning must be actively desired and controlled by the learner, (2) learners must be able to see the immediate benefit of learning something new to their current working situation, (3) environments must be intrinsically motivating so users can achieve large effects with reasonably small efforts, and (4) environments must allow learners to develop a functioning artifact from a design idea in a short time, so they can spend more time on things they really want to do, invest less ego in a particular product, and therefore be more willing to criticize it and change their designs.

3 Conceptual Frameworks

Self-Directed Learning. By putting the choice of tasks and goals under the control of the learner, self-directed learning can contribute to the goal that learning should simply be a natural consequence of being alive and in touch with the world rather than a process separate from the rest of life. It provides and exploits opportunities and support for helping people take charge of their own learning. By being the owners of problems, humans engaged in self-directed learning are able to integrate problem framing and problem solving, rather than just solve given problems.

Integration of Problem Framing and Problem Solving. The hardest and most important aspect of real world problem solving is not to solve a *given* problem but to figure out *what problem to solve*. The strong intertwining between problem framing and problem solving denies the objective existence of problems [Lave 88], it focuses efforts to search *for* a problem space rather than just *within* a problem space. It emphasizes the importance of putting problem owners in charge, because they have the authority and the knowledge to redefine the problem on the fly. By solving their own problems in self-directed learning, learners are problem owners who have the opportunity and face the challenge to frame and reframe problems.

Making Information Relevant to the Task at Hand. High-functionality systems confront us with too much information. The challenge is to make the information relevant to the task at hand—that is, delivering the right knowledge, in the context of a problem or a task, at the right moment for a human professional to consider [Nakakoji, Fischer 94]. A shared understanding is required by the system to achieve these objectives.

Integrating Knowledge in the Head and Knowledge in the World. Learning, understanding, and working in the real world rely heavily on integrating knowledge in the head and knowledge in the world [Lave 88; Norman 93]. Rather than emphasizing closed-book exams and tool-free performance, cognitive artifacts are used to expand people's mental power. There is less insistence on symbolic performances. Instead, actions are intimately connected with objects and events in the world, and people use the objects themselves rather than abstractions in their reasoning.

4 Examples of Learning Opportunities Supported by DoDEs

DoDEs [Fischer et al. 91; Eisenberg, Fischer 94] contain

- *different components* — (1) a construction component to create artifacts; (2) an argumentation component to explore and document the rationale behind an artifact; (3) a catalog of cases (i.e. previously developed artifacts); (4) a specification component for framing a problem; and (5) a simulation component to understand the relationship between structural and functional properties, and

- *integration mechanisms* — (1) computational critics (identifying breakdowns [Fischer 94], which might have remained unnoticed without them; these breakdowns provide learning opportunities for designers by supporting them to reframe problems, to attempt alternative design solutions, and to explore relevant background knowledge); (2) an argumentation illustrator (contextualizing general arguments with specific examples drawn from the catalog); and (3) a case deliverer (selecting the most relevant examples with respect to a given partial specification and construction from the catalog).

We have developed a domain-independent multifacted architecture underlying all of our design environments. By populating all the components and integration mechanisms with domain-specific knowledge, we have created a variety of different DODEs over the last years, including environments for kitchen design, computer network design, graphic arts design, and lunar habitat design.

DODEs provide multiple learning opportunities and learners have considerable control using the environments (e.g., they can disable the critiquing mechanisms, and they can ignore or explore the argumentation). Design environments, rather than being only passive tools or repositories of information, have evolved into agent-like systems cooperating with learners in searching, interpreting and creating knowledge [Nakakoji, Fischer 94].

An Example: The Voice Dialog Design Environment (VDDE). Voice dialog interfaces (see Figure 3; [Repenning, Sumner 92]) consist of a series of voice prompted menus. Users press buttons on a telephone keypad and the system responds with appropriate voice instructions. Current interface design techniques for voice dialog systems are based on flow charts. It is difficult for designers, customers and end-users of these systems to anticipate what the (audio) interaction will sound by simply looking at a static visual diagram. To experience breakdowns, simulations are needed which can serve as representations for mutual understanding by allowing designers, customers and end-users "experience" the actual audio interface. VDDE-Critics [Harstad 93] add critics to VDDE to signal additional breakdowns for the designers.

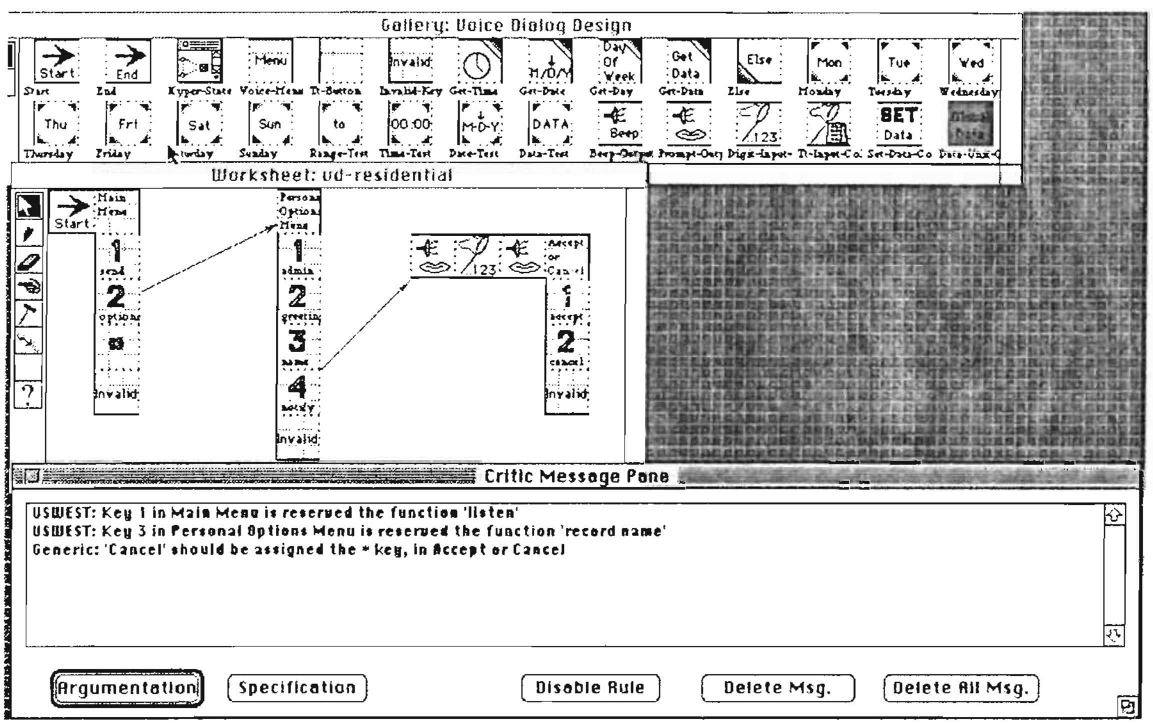

Figure 3: The Voice Dialog Design Environment (VDDE)

The VDDE allows domain designers to create graphic specifications. The top window is a gallery of domain-oriented components. The middle window is a worksheet where designers create a specific design. The behavior of the design can be simulated at any time. Design simulation consists of a visual trace of the execution path combined with audio feedback of all prompts and messages encountered. The bottom window displays the critiquing message for the design under construction.

5 Assessment

An Assessment of Our Efforts Based on New Cognitive Science Approaches Toward Learning. Current cognitive theories emphasize that learning is a process of knowledge construction, not one of knowledge recording or absorption [Resnick 89]. In our environments, this aspect is supported by the active engagement of the learner. Learning is knowledge dependent; people use current knowledge to construct new knowledge and to restructure existing knowledge. This aspect is supported by the argumentation and catalog and the access mechanisms provided by the critiquing component and the case deliverer.

Learning is also highly tuned to the situation in which it takes place [Lave 88]. No amount of knowledge of principles suffices to account for or guarantee the success of action in real-world problem situations. This challenges many of the basic assumptions about learning general problem-solving skills in a decontextualized way. Contextual elaboration is needed to devise specific courses of action in order to go beyond general procedural prescriptions.

Challenges in Creating Instrumental Computational Environments in Support of Self-Directed Learning. System-building efforts supporting self-directed learning face the challenge of creating systems that (1) can relinquish control of task selection yet maintain knowledge of users' goals, plans, and background knowledge; and (2) can function effectively in large solution spaces. Our work addresses these problems by representing specific problem domains with DoDEs. A partial understanding of the task at hand (as captured by the domain-orientation, by a specification and a construction) allows the system to obtain a partial shared understanding of the users' goals and intention, and to prioritize information spaces in support of self-directed learning.

Evaluation. Our work has proceeded as cycles of "design—assessment/evaluation—redesign." As mentioned before, the evaluation of our early critiquing systems [Fischer et al. 91] led to the need for embedding them in design environments. The recognition that design is an argumentative process [Schoen 83] demonstrated the need to back up the critiquing messages with argumentation. Argumentation itself needed to be contextualized through examples. The growing amount of information in our design environments made browsing infeasible and required components for information delivery and access.

Acknowledgments. The author would like to thank the members of the Human-Computer Communication group at the University of Colorado who contributed substantially to the conceptual framework and the systems discussed in this paper. The research was supported by (1) the National Science Foundation under grant MDR-9253245, (2) NYNEX Science and Technology Center (White Plains, N.Y.), and (3) Software Research Associates, Inc. (Tokyo, Japan).

References

[Boecker, Eden, Fischer 91]
H.-D. Boecker, H. Eden, G. Fischer, *Interactive Problem Solving Using Logo*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1991.

[Carroll, Rosson 87]
J.M. Carroll, M.B. Rosson, *Paradox of the Active User*, in J.M. Carroll (ed.), *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, The MIT Press, Cambridge, MA, 1987, pp. 80-111, ch. 5.

[Csikszentmihalyi 90]
M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*, HarperCollins Publishers, 1990.

[Eisenberg, Fischer 94]
M. Eisenberg, G. Fischer, *Programmable Design Environments: Integrating End-User Programming with Domain-Oriented Assistance*, Human Factors in Computing Systems, CHI'94 Conference Proceedings (Boston, MA), ACM, 1994, pp. 431-437.

[Fischer 91]
G. Fischer, *Supporting Learning on Demand with Design Environments*, Proceedings of the International Conference on the Learning Sciences 1991 (Evanston, IL), Lawrence Birnbaum (ed.), Association for the Advancement of Computing in Education, Charlottesville, VA, August 1991, pp. 165-172.

[Fischer 94]
G. Fischer, *Turning Breakdowns into Opportunities for Creativity*, Knowledge-Based Systems, Special Issue

on Creativity and Cognition, 1994, (forthcoming).

[Fischer et al. 91]
G. Fischer, A.C. Lemke, T. Mastaglio, A. Morch, *The Role of Critiquing in Cooperative Problem Solving*, ACM Transactions on Information Systems, Vol. 9, No. 2, 1991, pp. 123-151.

[Harstad 93]
B. Harstad, *New Approaches for Critiquing Systems: Pluralistic Critiquing, Consistency Critiquing, and Mulitple Intervention Strategies*, Technical Report CU-CS-685-93, Department of Computer Science, University of Colorado, 1993.

[Lave 88]
J. Lave, *Cognition in Practice*, Cambridge University Press, Cambridge, UK, 1988.

[Nakakoji, Fischer 94]
K. Nakakoji, G. Fischer, *Intertwining Knowledge Delivery and Elicitation: A Process Model for Human-Computer Collaboration in Design*, Knowledge-Based Systems Journal, 1994, (in press).

[Norman 93]
D.A. Norman, *Things That Make Us Smart*, Addison-Wesley Publishing Company, Reading, MA, 1993.

[Papert 80]
S. Papert, *Mindstorms: Children, Computers and Powerful Ideas*, Basic Books, New York, 1980.

[Polson, Richardson 88]
M.C. Polson, J.J. Richardson (eds.), *Foundations of Intelligent Tutoring Systems*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1988.

[Repenning, Sumner 92]
A. Repenning, T. Sumner, *Using Agentsheets to Create a Voice Dialog Design Environment*, Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing, ACM Press, 1992, pp. 1199-1207.

[Resnick 89]
L.B. Resnick, *Introduction*, in L.B. Resnick (ed.), *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1989, pp. 1-24, ch. 1.

[Schoen 83]
D.A. Schoen, *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, New York, 1983.

[Wenger 87]
E. Wenger, *Artificial Intelligence and Tutoring Systems*, Morgan Kaufmann Publishers, Los Altos, CA, 1987.