

Distributed Cognition, Learning Webs, and Domain-Oriented Design Environments

Gerhard Fischer

*Center for LifeLong Learning and Design (L³D)
Department of Computer Science and Institute of Cognitive Science
University of Colorado at Boulder*

Abstract

The human mind is limited, making collaboration with other humans and with things (in our case with computational environments) a necessity rather than a luxury. Relevant knowledge for work and for learning is distributed in our head, in the heads of others, and in the environment.

Learning webs are used by (virtual) communities of practice. Domain-oriented design environments (DODEs) support learning webs by allowing all stakeholders in a design process to learn and work collaboratively with each other and with their computational environments. DODEs serve as models for the design of collaborative working and learning environments by exploring and supporting different relationships and task responsibilities between humans and computers.

DODEs integrate working and learning by grounding learning in self-directed, authentic activities. They support learning on demand as an essential element of life-long learning. The creation of DODEs faces the fundamental challenge to make them simultaneously learner-directed and supportive. DODEs transcend other computer-supported cooperative learning systems, which employ the computer only as a medium with few interpretable components. They integrate humans and computational resources more creatively by acknowledging that persons become skill resources only when they consent to do so, whereas computational environments are available at the bidding of the user.

Keywords — situated learning in the workplace, environments for open-ended and termless learning, theories of collaboration and learning, distributed cognition, life-long learning, learning on demand, learning webs, domain-oriented design environments.

1. Introduction

In our research over the last fifteen years, we have created conceptual frameworks and innovative systems and we have conducted assessment studies to address

problems of working, learning, and collaboration with computational artifacts. The content domains of our work are design activities in which design is understood very broadly as the process of determining how things ought to be (Simon 1981). Design can be seen as a fundamental activity within all professions. It is a collaborative, argumentative process without optimal solutions but with trade-offs. It is impossible for design processes to account for every aspect that might affect the designed artifact. Therefore, design must be treated as an evolutionary process, in which all stakeholders continue to learn new information and insights as the process unfolds (Fischer, McCall et al. 1994).

The necessity to intertwine learning, working, and collaboration results from the growing recognition that in the information age, change is unavoidable and obsolescence is guaranteed. Learning can no longer be considered a process that occurs only in schools. We have to think of learners not as being inherently isolated but rather as having to learn to make new, different, and strategic uses of the sources of information around them. The successful student or professional is one who learns how to use research materials, libraries, and computational environments, as well as knowledgeable humans (parents, teachers, peers, mentors, and practitioners from other disciplines) to master complex problems.

2. Distributed Cognition: Limitations of the Individual, Unaided Human Mind

2.1. Limitations of the Individual Human Mind

Human beings have a bounded rationality (Simon 1981). There is only so much we can remember and there is only so much we can learn. Talented people require approximately a decade to reach top professional proficiency. These general observations provide the rationale that, when a domain reaches a point where the knowledge for skillful professional practice cannot be acquired in a decade, specialization will increase, teamwork will become a necessity, and practi-

tioners will make increasing use of external reference aids (such as printed and computational media). With powerful technologies becoming widely available, people take on more complex jobs. Therefore, they need help in accomplishing unfamiliar tasks that are part of an expanded job. Beyond the need for new and changing domain knowledge, there is also a large demand for new tool knowledge.

2.2. Distributed Cognition

Learning is 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. Acquiring knowledge cannot be restricted to obtaining a prescribed education at a given time. 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). Education should be a distributed lifelong process by which one learns material as one needs it. Distributed cognition (Norman 1993) is a necessity in response to the limitations of the individual human mind.

Distributed cognition needs to include humans and things, and the two infrastructures should complement each other. Humans (e.g., coaches, peers, practitioners from other domains) have extensive background knowledge and a shared understanding unavailable in things. Things can store information (e.g., books), highlight relevant information (e.g., graphs, mathematical notations), and retrieve, compute, and analyze information (e.g., different forms of computational media).

3. Learning Webs

Illich (Illich 1971) (long before the world-wide web and the information superhighway were a reality) has envisioned learning webs as an alternative and augmentation to traditional schooling. The major objectives that he envisioned his learning webs would provide were (1) reference services to educational objects, (2) skill exchanges, (3) peer matching, and (4) reference services to educators-at-large. Many collaboration technologies (e.g., most Computer-supported collaborative work systems) employ the computer as a medium with few interpretable components. Future computational environments need to integrate humans and computational resources more creatively. Computational environments that can interpret objects, actions, and artifacts (not only from a tool perspective but also from a domain perspective) can make information and resources available at the bidding of the user, whereas persons become skill resources only when they consent to do so, and they can also restrict time, place, and methods as they choose.

To increase the computational support of collaborative environments, a limited shared context must be established. General-purpose information spaces can

have only a limited notion of users' tasks at hand. Domain-oriented design environments (DODEs) (Fischer 1994) exploit domain semantics and the design context to actively notify designers when there is information they should know. Many current design systems are limited because they function only as "keepers" of the artifact, in which one deposits representations of the artifact being designed. Our experience has shown that designers integrate designing and discussing in such a way as to make separate interpretation difficult (Reeves 1993).

4. Domain-Oriented Design Environments

DODEs have emerged in our research work as computational environments in support of collaboration. They are semiformal systems that integrate object-oriented hierarchies of domain objects, rule-based critiquing systems, case-based catalog components, simulation components, checklists, and argumentative hypermedia systems. They support communications and negotiations among all involved stakeholders and between the designers and their work in progress. They do limited reasoning and interpretations, trigger breakdowns, deliver information, and support the exploration of the rationale behind the artifact.

The goals associated with DODEs are (1) to bring task to the forefront by supporting human problem-domain interaction, (2) to create a shared context between designers and computational environments, (3) to create an artifact-centered information repository facilitating collaboration among all stakeholders, (4) to support learning on demand and information delivery, and (5) to have human designers in control. The theories underlying DODEs are (1) to make objects and ideas ready-at-hand, allowing learners to communicate more directly with the task, (2) to support reflection-in-action (Schön 1983), (3) to integrate problem framing and problem solving, (4) to allow design-in-use, and (5) to increase the back-talk of the situations (Fischer, Lemke et al. 1991). The users of DODEs are skilled domain workers who belong to the community of practice that a specific DODE supports.

4.1. An Example: The Voice Dialog Design Environment

The Voice Dialog Design Environment (VDDE) (Repenning and Sumner 1992) will be used to illustrate our conceptual framework. Voice dialog interfaces 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 how the (audio) interaction will sound by simply looking at a static visual diagram. To experience breakdowns, simulations are needed that can serve as representations for mutual understanding by allowing

designers, customers, and end-users to “experience” the actual audio interface.

The VDDE allows domain designers to create graphic specifications using a gallery of domain-oriented components and worksheets on which 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.

Earlier versions of VDDE did not contain a critiquing component, limiting the “back-talk” to the designers and the learning opportunities provided for them. Voice dialog design is complicated by the fact that there are different rule sets that should be obeyed by a design. VDDE-Critics (Harstad 1993) adds critics to VDDE to signal additional breakdowns for the designers. In addition to earlier critiquing systems, VDDE-Critics allows designers to tailor the “breakdown” characteristics of the system to their personal needs by (1) selecting the rule set and the associated argumentation to be used, (2) determining the intrusiveness of the critiquing mechanisms with the critiquing thermometer, and (3) choosing the design component to be critiqued (a conceptual unit versus the overall design).

4.2. A Process Model Illustrating Collaborative Processes in Design Environments

Design problems are intrinsically ill-defined, open-ended, and “wicked,” making it impossible to predict, let alone collect, all the potentially relevant information in advance. They must capture information continuously over the lifetime of the system and make that information available to designers when it is relevant to their particular tasks. We have developed the SER model, a process model for the evolution of domain-oriented design environments (Fischer, McCall et al. 1994) consisting of three phases: seeding, evolutionary growth, and reseeded.

During seeding, environment developers and domain designers collaborate to create a design environment seed. During evolutionary growth, domain designers create artifacts that add new domain knowledge to the seed (i.e., new knowledge is generated and integrated into the environment by the domain designers themselves rather than produced by the environment developers). In the reseeded phase, environment developers again collaborate with domain designers to organize, formalize, and generalize new knowledge.

4.3. Increasing the Situation Awareness

Design is a well-suited activity to explore concepts in collaboration because the design activity takes place *within* the computational environment. The “situation awareness” of a DODE is increased through the following mechanisms: (1) the domain orientation allows a default intent to be assumed, namely, the creation of an artifact in the given domain; (2) the construction

situation is accessible and can be “parsed” by the system, providing the system with information about the artifact under construction; (3) the specification component allows one to explicitly communicate high-level design intentions to the system; and (4) the embedding of annotations contextualizes messages to other stakeholders rather than communicating them in a decontextualized e-mail message.

4.4. Learning on Demand and End-User Modifiability

DODEs provide learning-on-demand opportunities (Fischer, Lemke et al. 1991) for a designer through critiquing, simulation, and access to contextualized argumentation and cases. But the information flow is not only one-directional. Using DODEs, designers will transcend the existing knowledge and contribute new knowledge themselves. Because these designers are domain designers and not software designers, end-user modifiability support is required.

End-users may wish to have functionality that fits their needs, but the creation of this functionality is a difficult task. Two major approaches, namely programmable design environments (Eisenberg and Fischer 1994) and collaborative work practices (Nardi 1993), make end-user programming a more realistic challenge. Collaborative work practices, leading to the development of power users and local developers, are naturally developing practices in communities where end-user modifiable tools are available.

5. The Support of DODEs for Collaborative Learning—Lessons Learned From Our System-Building Efforts

If things are basic resources for learning, then the quality of the environment and the relationship of persons to them will determine how much they will be able to learn. DODEs are instrumental versions of systems that are *simultaneously* user-directed and computationally supportive (thereby complementing open learning environments and intelligent tutoring systems with an additional alternative). DODEs support human problem-domain communication by reducing the demands of learning about the tool. They offer a variety of different learning opportunities through critiquing, simulation, argumentation, and examples. Having an increased situational awareness through the integration of the different components, DODEs are able to incrementally obtain a partial understanding of the task at hand and to contextualize information to it.

While cognitive questions and content are important, collaboration technologies raise numerous other issues. What will make people want to share? What will motivate people to make their knowledge explicit and contribute it to an organizational memory (especially, if they have to do the work but are not nec-

essarily the beneficiaries of it)? These questions will in the long run be more important than technological issues, and successful models to answer them positively are still quite rare.

One of the benefits of integrating working and learning is the potential increase in motivation. Motivation to learn new things is critically influenced by optimal flow, a continual feeling of challenge, direct engagement, the right tools for the job, and a focus on the task (Csikszentmihalyi 1990). Users are willing and motivated to learn when the following conditions hold:

- (1) They actively desire and control learning—supported in DODEs by the integration of working and learning allowing learners to be engaged in authentic, self-directed activities.
- (2) They are successful in finding and using new information—supported in DODEs by contextualizing new information to the task at hand and to breakdown situations.
- (3) They can see the immediate benefit of learning something new to their current working situation—supported in DODEs by making argumentation serve design, by locating relevant catalog examples, and by illustrating complex behavior with simulations.
- (4) Their environments are intrinsically motivating and allow them to achieve interesting results with a reasonably small effort—supported in DODEs through human problem domain communication, which allows users to focus on their tasks.

6. Conclusions

Design activities require learning and collaboration. We have developed conceptual frameworks and innovative systems that support not only the creation of the artifact but also the professional communities engaged in design as professional practitioners. We have learned from our efforts that older frameworks of education—associated with notions of instructionism, memorization, and decontextualized learning—cannot be shaken merely by the presence of technology, whether that technology takes the form of intelligent tutoring systems, multimedia, or world-wide connectivity. New frameworks must instead be devised to support lifelong learning—learning webs that allow integration of learning, working, and collaborating; engagement in authentic problems; self-direction in learning tasks; and creation of new content and domain areas.

Acknowledgments

The author would like to thank the members of the “Center for LifeLong Learning and Design” at the University of Colorado. The research was supported by (1) the National Science Foundation, Grant RED RED-9253425; (2) the ARPA HCI program, Grant N66001-94-C-6038; (3) the National Science Foundation and the Advanced Research Projects Agency under Cooperative Agreement No. CDA-940860; (4) NYNEX Science and Technology Center; (5) Software Research Associates (SRA); and (6) PFU. During academic year 1994/95, the author was supported by the “SEL-Stiftungsprofessur” of the Technical University Darmstadt.

References

- Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience*. HarperCollins Publishers.
- Eisenberg, M. and G. Fischer (1994). *Programmable Design Environments: Integrating End-User Programming with Domain-Oriented Assistance*. Human Factors in Computing Systems, CHI'94 Conference Proceedings. Boston, MA, 431-437.
- Fischer, G. (1994). *Domain-Oriented Design Environments. Automated Software Engineering*. Boston, MA, Kluwer Academic Publishers. 177-203.
- Fischer, G., A. C. Lemke, et al. (1991). The Role of Critiquing in Cooperative Problem Solving. *ACM Transactions on Information Systems*. 123-151.
- Fischer, G., R. McCall, et al. (1994). Seeding, Evolutionary Growth and Reseeding: Supporting Incremental Development of Design Environments. Human Factors in Computing Systems, CHI'94 Conference Proceedings (Boston, MA). 292-298.
- Harstad, B. (1993). *New Approaches for Critiquing Systems: Pluralistic Critiquing, Consistency Critiquing, and Multiple Intervention Strategies*, Master Thesis, Department of Computer Science, University of Colorado, Boulder
- Illich, I. (1971). *Deschooling Society*. New York, Harper and Row.
- Nardi, B. A. (1993). *A Small Matter of Programming*. Cambridge, MA, The MIT Press.

- Norman, D. A. (1993). *Things That Make Us Smart*. Reading, MA, Addison-Wesley Publishing Company.
- Reeves, B. N. (1993). *The Role of Embedded Communication and Artifact History in Collaborative Design*. University of Colorado.
- Repenning, A. and T. Sumner (1992). Using Agentsheets to Create a Voice Dialog Design Environment. *Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing*. ACM Press. 1199-1207.
- Schön, D. A. (1983). *The Reflective Practitioner: How Professionals Think in Action*. New York, Basic Books.
- Simon, H. A. (1981). *The Sciences of the Artificial*. Cambridge, MA, The MIT Press.