

# Boundary Objects: Their Role in Articulating the Task at Hand and Making Information Relevant to It

Ernesto G. Arias and Gerhard Fischer

Center for LifeLong Learning & Design and Institute of Cognitive Science  
Department of Computer Science, and College of Architecture and Planning  
University of Colorado, Boulder

## Abstract

Complex collaborative design activities cannot be solved by individuals or by single groups. Communities of interest (defined by their collective concern with the resolution of a problem) bring together different communities of practice which represent groups of practitioners from different domains. Reaching common understanding between these communities is a major challenge for information technologies due to the communication divide produced by their respective cultures [Snow, 1993]. Therefore, future HCI innovations need to include notions such as “boundary objects,” as necessary notions if they are to effectively support communication between such communities. The relevance of this notion is that without it such these communities cannot create the necessary shared understanding behind the framing and resolution of the emerging problem in design activities. Nor, we further argue that without it, can all stakeholders be empowered to actively contribute and critique constructively the contributions from other communities of practice [Arias et al., 2000].

In this paper we develop a conceptual framework for collaborative design focused on boundary objects. This framework has been used in our theory-guided system development effort such as domain-oriented design environments and the Envisionment and Discovery Collaboratory to illustrate the need, the use, and the possibilities of creating design situation that can be understood and further evolved by all stakeholders.

## Keywords

boundary objects, collaborative design activities, computer mediated communication, computer supported collaborative learning, augmented reality, next generation of human-computer interaction systems, communities of interest, communities of practice

## Introduction

**The Two (or Many) Cultures.** C.P. Snow [Snow, 1993] has identified in his book two cultures (in his case: the literary intellectuals and the natural scientists) “who had

almost ceased to communicate at all.” He further claimed that “there exists a profound mutual suspicion and incomprehension, which in turn has damaging consequences for the prospects of applying technology to the alleviation of the world’s problems” and “there seems to be no place where the cultures can meet.” Over the years, we have brought together many different kinds of cultures such as people from academia and from industry, software designers and software users, and students and researchers from different countries and disciplines from around the world in our Center for LifeLong Learning and Design (L3D).

In a world of increasing specialization where the Renaissance scholar is a person of the past, collaboration (particularly in design) and its effective support are a fundamental challenges. 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 practitioners will make increasing use of reference aids for supporting distributed cognition [Fischer, 1995]. Distributed cognition requires environments that create and define new role distributions between humans and computers. Most of what any individual “knows” today is not in her or his head, but is out in the world (e.g., in other human heads or embedded in media). Humans are tool-using, tool-making species; we also rely on culturally devised ways of thinking, learning, working and collaborating.

In our research we have particularly focused on design [Simon, 1996]. In this paper, we (1) analyze different design communities from our focus, (2) articulate a conceptual framework centered around our notion of boundary objects, a task at hand, and how information can be relevant to that task; and (3) describe environments that support these design communities and which instantiate our conceptual framework in specific settings.

## Design Communities

### Communities of Practice (CoPs)

Communities of practice are made-up of practitioners who work as a community in a certain domain doing the similar work (although within each community there are individual

with special expertise, such as power users and local developers [Nardi, 1993]). Examples of communities of practice are: architects, urban planners, research groups, software developers, and software users. In our past work, we have developed various types of domain-oriented design environments [Fischer, 1994] to support communities of practice through its domain-orientation which supports interaction at the level of the problem domain of the community of practice and not only at a computational level.

Sustained engagement and collaboration lead to boundaries [Wenger, 1998] which are based on shared histories of learning and create discontinuities between participants and non-participants. At L3D (which as a research group represents a CoP), we have created over the last fifteen years concepts, systems, and stories representing an efficient and effective means for communication within the research group. However, what is empowering to the insider is often a major hurdle for outsiders or for newcomers who join the group. To address these problems, we have created Dynagloss as a central glossary that is extensible by users, who can add new terms, annotate current definitions, and propose new definitions. It is a particular application of the DynaSites system [Ostwald, 2000] which in turn is an environment for creating and evolving collections of Web-based information spaces that are by nature open-ended and grow through the contributions of users.

### **Communities of Interest (CoIs)**

*Communities of interest* are groups similar to CoPs, but from different backgrounds coming together to solve a particular (design) problem of common concern. Examples of CoIs could be (1) a team interested in software development that includes software designers, marketing specialists, psychologists, and programmers, or (2) a group of citizens and experts interested in urban transportation planning who are concerned with implementing new transportation systems, as illustrated later on in the paper by the Envisionment and Discovery Collaboratory. CoIs are “defined” by their shared interest in the framing and resolution of a design problem. Fundamental challenges facing CoIs are found in building a shared understanding of the task at hand (which often does not exist upfront, but is evolved incrementally and collaboratively and emerges in people’s mind and in external artifacts). Members of CoIs need to learn to communicate with and learn from others who have a different perspective and perhaps a different vocabulary for describing their ideas and establish a common ground and a shared understanding [Arias, 1996].

## **The Conceptual Framework and Theory**

### **Symmetry of Ignorance**

Design activities, given the nature of their context (wicked problems, change, on-demand, multiple stakeholders) often evolve over long periods of time. Complexity in design arises from the need to synthesize different perspectives of stakeholders on a problem, the management of large amounts of information relevant to a design task, and understanding the design decisions that have determined the long-term evolution of a designed artifact. Further, the knowledge associated with a design problem is tacitly distributed tacitly among the various stakeholders as owners of the problem. Given such context, design activities require collaboration among different stakeholders and therefore is characterized by a *symmetry of ignorance* [Rittel, 1984]. This characterization means that no individual stakeholder, or group of stakeholders, such as a CoP, knows all the relevant knowledge, yet the knowledge of all of them is equally (symmetrically) important in the process of framing and resolving the problem. Communication among stakeholders is difficult as mentioned because they use different languages for external cognition. Therefore, to provide a support context for communication, explicit representations to ground collaborative design activities are needed [Arias, 1996].

### **Articulating the Task at Hand**

An important outcome of shared understanding among CoPs and CoIs is the incremental creation of externalizations [Bruner, 1996] to capture and articulate the *task at hand* [Fischer et al., 1995]. Information is relevant to the task at hand if it (1) helps all participating stakeholders to understand a problem, and (2) is made available when the need for it arises. Externalizations enhance mutual understanding and intelligibility by serving as a resource for assessing the relevance of information within the context of collaboration. In everyday communication between people, externalizations are often communicated against a rich background of shared experience which is often available only in a very limited form among members of a community. In addition, stakeholders do not like to study large information repositories in the abstract (such as many pages of design rationale, of user manuals, etc.). They are interested in aspects of situations that are directly relevant to their goals and objectives and which help them understand problematic aspects of the design situation.

### **Boundary Objects**

Boundary objects are objects that serve to communicate and coordinate the perspectives of various constituencies, e.g., different CoPs brought together for some purpose [Wenger, 1998] thereby forming a CoI. In everyday life we constantly deal with artifacts that connect us in various ways to CoPs

to which we do not belong. Boundary objects serve multiple constituencies in situations where each constituency has only partial knowledge (based on the symmetry of ignorance) and partial control over the interpretation of the object. In this manner, boundary objects perform a brokering role [Fischer & Reeves, 1995] involving translation, coordination and alignment between the perspectives of specific CoPs. It must be understood that the efficiency of the boundary objects in attaining these functions is also contingent on the nature of the constituencies (e.g., their respective level of competency, motivation, and experience).

Externalizations often serve the purpose to create “situations that talk back to us” [Schön, 1983]. This “back-talk” will be severely limited by representations that do not serve as boundary objects. While some of the back-talk will be provided by the design situation itself, this may be insufficient because our ability to notice breakdowns and problematic situations by (visual) inspection and careful analysis is limited. In our research over the last decade we have developed additional mechanisms to further increase the “back-talk” [Fischer, 1994]: (1) feedback from human stakeholders involved in the design process, (2) computational critics, and (3) simulation components that illustrate the behavior of an artifact. In providing additional feedback, it is important that the “back-talk” is relevant to the actual design situation and that it is articulated in a way that the designer can understand. In the construction of shared understanding, the mutually complementary functions behind boundary objects include exploiting the power of the symmetry of ignorance by making the tacit explicit, as well as utilizing the asymmetry of knowledge (knowledge of stakeholders, while equally relevant is different) by eliciting the relevant at the appropriate time.

### **Boundary Objects — Bridging the Gap between Situation and System Models**

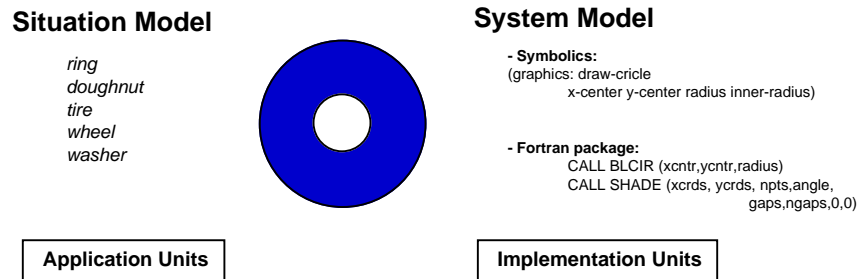
Boundary objects as described, can serve two major purposes: (1) they can serve as objects to support the interaction and collaboration between different communities of practice, and (2) they can serve the

interaction between users and (computational) environments. In this later case one can argue that they serve the interaction between the users and the designers (being present “virtually” through the system created by them).

The interaction between a user and a system is a conversation in a vocabulary and language determined by the input the system is able to accept and process. Many systems (e.g., help systems, documentation systems) are oriented toward the system rather than toward the user: information is structured around a description of the system, not necessarily around an analysis of the problems users address when using the system.

Compared to humans, most computer systems are able to respond to a very limited range of input and users must provide input to the system that is within the limited range to which the computer is programmed to respond. We have used the terms *situation* and *system models* to talk about the discrepancies between the way a human thinks of a problem and the limited inputs to which a system is capable of responding. The *situation model* is a mental representation of the situation as the user sees it, including the problems motivating a task, and general ideas for finding a solution [Kintsch, 1998]. The *system model* consists of a set of operations that, when invoked, will result in the desired (programmed) solution. These operations must all be within the repertory of the system; that is, for each operation there must exist one or more commands, depending upon context, to execute it. At the level of the situation model, goals refer to actions and states in the user’s problem space and are articulated in terms of what we want. Goals may be precise or imprecise, but the important point is that they are not necessarily structured or named according to the system model [Furnas et al., 1987].

When users approach a problem, they often begin at a high level of abstraction, conceptualizing the design in terms of the application problem to be solved [Curtis et al., 1988]. This initial conceptualization must then be translated into terms that the computer can understand. For example: let’s



**Figure 1: The Gap between Situation and System Models**

assume that users wish to draw a ring-like figure (as shown in Figure 1) on a Symbolics Lisp Machine [Fischer et al., 1991]. To do so, they must know the system model, which treats this object as the “inner-radius” option to the “draw-circle” function. In the situation model, objects are conceptualized in terms of application units, such as “ring” or “donut.” In the system model, objects are represented using implementation units, as illustrated by the Symbolics and Fortran representations.

There are a number of possibilities to develop and use boundary objects to reduce the gap between situation and system models:

- development of a *new system model* which is closer to an individual’s situation model and hence more intuitive and easier to use [Fischer, 1994]. This is the goal of human problem-domain interaction – to present a system model to users that closely matches their situation model of the domain.
- support for users to bring their situation model closer to the system model by making the relevant features of the latter more transparent. An example of this approach is the paradigm of retrieval by reformulation [Williams, 1984; Fischer & Nieper-Lemke, 1989]. By presenting examples of objects found by queries, the system reveals its system model and helps the user to incrementally formalize their query to better match the system model.
- translation of a request by a knowledge-based agent [Fischer & Reeves, 1995] from the situation model into the system model. In this case, the agent must contain enough knowledge to assist users in mapping tasks conceptualized in their situation model to the system model.
- training of users to express themselves in the system model. This is the case with many skilled expert users, who have gained over time a familiarity with the system model such that their situation model matches the system model. This has also been our experience in non-computational system models such as physical games in urban planning [Arias, 1996].

Current support information is system-oriented rather than user-oriented. Documentation and help information is structured to describe the system, not to address the problems experienced by the user. This is why human assistance, if available on a personal level, is still the most useful and used source of advice and help. Learners can ask a knowledgeable colleague a question in an infinite variety of ways; they get assistance in formulating the question, and they can articulate their problem in terms of the situation model. A knowledge store trying to cover situation models must incorporate “user constructs”, user-oriented organizations of knowledge, and a presentation component that presents information in the user’s concepts and words.

## Innovative Systems behind our Conceptual Framework

### Supporting Boundary Objects in Domain-Oriented Design Environments (DODEs)

Design problems, being ill-defined and unique, require informed participation by all stakeholders [Arias et al., 1999]. Openness and complexity in design arises from the need to synthesize different perspectives of a problem, the creation of *boundary objects* [Resnick et al., 1991], the management of large amounts of information relevant to a design task, and understanding the design decisions that have determined the long-term evolution of a designed artifact.

One such challenge is to integrate the various perspectives emerging from the symmetry of ignorance among articulate stakeholders. By supporting the process of reflection within a shared context defined by the task at hand, opportunities can emerge from enhancing the creation of shared understanding. This process melds the information that is collaboratively constructed into the problem-solving context, informing the process as well as the stakeholders and allowing them to participate from a more enriched and meaningful perspective.

The “situation awareness” of a DODE [Fischer, 1994] 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 the user 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 as decontextualized e-mail messages).

### The Environment and Discovery Collaboratory (EDC) — Next Generation Design Environments

The Environment and Discovery Collaboratory (EDC) (<http://www.cs.colorado.edu/~13d/systems/EDC/>) [Arias et al., 2000a] is a domain-oriented design environment under development to support social interaction by creating *shared understanding* among various stakeholders, *contextualizing* information to the task at hand, and creating *boundary objects* as externalizations in collaborative design activities. The EDC framework is applicable to different domains, but our initial effort has focused on the domain of locational decision-making in urban planning, specifically in transportation planning and community development. Creating shared understanding requires a culture in which stakeholders see themselves as reflective practitioners rather than all-knowing experts [Schön, 1983]. The symmetry of ignorance is a defining notion of such collaborative design activities: stakeholders

are aware that while they each possess equally important knowledge, none of them has all the relevant knowledge to frame or resolve such problems.

A central theoretical vision of the EDC is to provide contextualized support for reflection-in-action [Schön, 1983] within collaborative design activities. Figure 2 shows the current realization of the EDC environment. Using the horizontal electronic whiteboard, participants work “around the table” creating incrementally a shared model of the problem. They interact with computer simulations in the action space by manipulating three-dimensional, physical objects, which constitute a language for the domain. The position and movement of these physical objects are recognized by means of the touch-sensitive projection surface. In the figure, users are constructing a neighborhood through the use of a physical language appropriate for the problem by placing objects representing houses, cars, traffic lights, and so on. This construction then becomes the object through which the stakeholders can collaboratively evaluate and prescribe changes in their efforts to frame and resolve a problem [Arias, 1996]. In the upper half of Figure 2 is a second electronic whiteboard that serves as the *reflection space*, where information related to the problem-at-hand can be presented, explored, and reframed. In the figure a user is filling out a survey constructed from the model presented in the action space. The results of this survey are stored in the reflection space (for future exploration) and are also fed to the action space, where the ramifications of the decisions specified in the survey can be explored or utilized in the construction of meaning behind the language of objects.



Figure 2: The EDC Environment

Much past and current development of technology for design builds on, or is constrained by the “single-user / single-computer” interaction model. The EDC emphasizes the creation of shared interaction and the cultural embedding for learning and design within the context of communities of learners. Crucial processes relevant for social interaction that are supported by the EDC are its ability to:

- deal with a set of possible worlds effectively (i.e., support exploration of design alternatives) to account for the fact that design is an argumentative process in which the goal is not to prove a point but instead to create an environment for a design dialog;
- incorporate an emerging design in a set of external memory structures, and record the design process and the design rationale;
- generate low-cost, modifiable models that assist stakeholders in creating shared understanding by engaging in a “conversation with the materials”;
- use simulations to engage in “what-if” games and to replace anticipation of the consequences of our assumptions by analysis;
- introduce the notion of a common language of design by integrating physical objects with virtual objects [Arias et al., 1997];
- allow the collaborative and incremental creation of boundary objects that serve as objects for mutual understanding for all participating stakeholders.

In this manner, the EDC is a contribution toward a new generation of collaborative systems. It shifts the emphasis away from the computer screen as the focal point and creates an *immersive* environment in which stakeholders can incrementally create shared understanding through collaborative design.

Utilizing novel computational and physical tools is a critical part of supporting this dynamic face-to-face interaction. Languages of physical objects provide affordances for the construction of shared, tangible representations that are used to frame problems in a collaborative manner [Alexander et al., 1977; Arias et al., 1997]. In both, the action and reflection spaces, creating an open seed that can evolve through use is an essential element of the EDC’s design. The computational model

is driven by AgentSheets, an open, user-extensible simulation environment [Repenning & Sumner, 1995]. The reflection space is built using DynaSites, a substrate for evolving and maintaining dynamic Web-based information spaces [Ostwald, 2000]. The action and reflection spaces are independent computational entities that utilize the Web as a communication medium. The EDC focuses on face-to-face collaboration when possible, but provides opportunity for distributed collaboration by allowing people to participate at a distance and by providing a persistent form of what takes place at individual meetings.

### Boundary Objects and the EDC

The vision behind the EDC is to shift the focus of future developments away from the computer towards an increased understanding of the human, social, and cultural system that defines the context in which systems are used. The EDC instantiates the conceptual frameworks and requirements outlined earlier and serves to create an *immersive* social context in which a community of stakeholders can create, integrate, and disseminate information relevant to their lives and the problems they face.

The EDC supports stakeholders in creating information by articulating their own knowledge in a form that other people can understand. The use of a shared physical context is one of the important ways to help people articulate their knowledge and communicate with others. Many have found integrated physical/computational systems to be interesting because of their power to elicit tacit communication [Arias et al., 1997; Eisenberg & Makay, 1996]. In the EDC, the physical representation serves as an external object through which users can express their views. It allows a group of neighbors to create an explicit understanding of what lies within their neighborhood. Through the common physical representation, users can utilize the language of objects to collaboratively determine what features are parts of their neighborhood.

The EDC supports knowledge dissemination by providing contextualized access to information. Rather than dissemination being the end of the process, the evolvable nature of the EDC supports a synergy between the dissemination of information and the creation of new

EDC Support	Description	Contribution to Different Aspects of Boundary Objects
physical languages	tangible physical representations that are manipulated by groups of users	encourages face-to-face collaboration, providing a common language for people to express themselves
computational simulations	models that capture constructions, analyze situations, and display ramifications	allows for users to engage in "what-if" games, provides interactive ways to capture and visualize information
dynamic web sites	evolving Web sites that display relevant information and capture	captures knowledge and helps make tacit knowledge explicit

	feedback	
integrated physical / virtual interaction	tight coupling between physical construction and reflective information	creates an engaging forum, motivates participation, helps deliver information in a timely contextualized manner
open evolvable tools (including end-user modifiability)	making modification a "first-class design activity", being able to capture changes and evolve systems when new situations arise	captures important information not anticipated at system design time, encourages a culture of participation, addresses the open-ended nature of problems

**Figure 3: EDC Support Mechanisms for Boundary Objects**

knowledge through informed participation [Arias et al., 1999]. The information can grow over time by allowing users to add their comments, new factual information, or related Web links associated with the issue. These capabilities of the EDC (as summarized in Figure 3) create a range of possibilities for the creation of boundary objects.

## Assessment

Creating boundaries objects is not without challenges. In many situations, the dual goal of creating an object understandable by every participating stakeholder and at the same time making an important contribution of a specific community of practice, is not easily achieved.

Although the physical representations used in the EDC have been used extensively in authentic real-world planning situations, the EDC as an integrated physical and computational environment is relatively new. The major contributions of the EDC have been to support participatory design in which expertise is distributed among different stakeholders, requiring that each stakeholder plays the role of both teacher and learner. To support mutual learning and shared understanding among different groups of stakeholders, representations such as boundary objects are needed which can be understood by all participants. The EDC as an immersive and emergent environment provides opportunities where different "cultures" can meet

If knowledge in the world is a basic resource for thinking, working learning, and collaborating, then the quality of the computational environments serving external cognition (and the relationship of users to them) will determine how much users will be able to profit from it. DODEs, and specifically the EDC, are instrumental versions of systems that are *simultaneously* user-directed and computationally supportive. Having an increased situational awareness through the integration of the different components, these environments simultaneously support (1) human-human collaboration with boundary objects, and (2) human-computer interaction by obtaining a partial understanding of the task at hand and to contextualize information to it.

## Conclusions

Without boundary objects and without making information relevant to the task at hand, communities of interest will be severely limited in their collaborative design activities. Domain-oriented design environments such as the Envisionment and Discovery Collaboratory provide a first step to exploit the powerful notion of boundary objects. Such notion, when properly utilized in when designing future HCI systems provides a basis to empower users in their collaborative design and learning by enhancing informed participation around the table and beyond.

## Acknowledgments

The authors would like to thank the members of the Center for LifeLong Learning & Design at the University of Colorado, who have made major contributions to the conceptual framework and systems described in this paper. We are especially grateful to Hal Eden, Andrew Gorman, and Eric Scharff, our co-developers of the EDC over the last five years. The research was supported by (1) the National Science Foundation, Grants REC-9631396 and IRI-9711951; (2) Software Research Associates, Tokyo, Japan; (3) PFU, Tokyo, Japan; and (4) the Coleman Foundation, San Jose, CA.

## References

- Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., & Angel, S. (1977) *A Pattern Language: Towns, Buildings, Construction*, Oxford University Press, New York.
- Arias, E. G. (1996) "Bottom-up Neighborhood Revitalization: A Language Approach for Participatory Decision Support," *Urban Studies*, Vol. 33, No. 10 pp. 1831-1848.
- G., Eden, H., & Fischer, G. (1997) "Enhancing Communication, Facilitating Shared Understanding, and Creating Better Artifacts by Integrating Physical and Computational Media for Design," *Proceedings of Designing Interactive Systems (DIS '97)*, pp. 1-12.

- Arias, E. G., Eden, H., Fischer, G., Gorman, A., & Scharff, E. (1999) "Beyond Access: Informed Participation and Empowerment," *Proceedings of the Computer Supported Collaborative Learning (CSCL '99) Conference*, pp. 20-32.
- Arias, E. G., Eden, H., Fischer, G., Gorman, A., & Scharff, E. (2000a) "Transcending the Individual Human Mind—Creating Shared Understanding through Collaborative Design," *Transactions on Computer Human Interaction*, (to appear). Available at: <http://www.cs.colorado.edu/~gerhard/papers/tochi99.pdf>.
- Arias, E. G., Fischer, E. G., Eden, H., Gorman, A., & Scharff, E. (2000b) "Enhancing Participation through Innovation: The Envisionment and Discovery Collaboratory," in G. Moser Metropolis 21st Century: Cities, Social Life and Sustainable Development, IAPS 2000, Universite Rene Descartes-Paris.
- Bruner, J. (1996) *The Culture of Education*, Harvard University Press, Cambridge, MA.
- Curtis, B., Krasner, H., & Iscoe, N. (1988) "A Field Study of the Software Design Process for Large Systems," *Communications of the ACM*, 31(11), pp. 1268-1287.
- Eisenberg, M. & Makay, W. (1996) "Real Meets Virtual: Blending Real-World Artifacts with Computational Media," *CHI '96, Human Factors in Computing Systems*, pp. 159-160.
- Fischer, G. (1994) "Domain-Oriented Design Environments," *Automated Software Engineering*, 1(2), pp. 177-203.
- Fischer, G. (1995) "Distributed Cognition, Learning Webs, and Domain-Oriented Design Environments." In J. L. Schnase & E. L. Cunnius (Eds.), *Proceedings of CSCL'95: The First International Conference on Computer Support for Collaborative Learning*, Lawrence Erlbaum Associates, Inc., Mahwah, NJ, pp. 125-129. Available at: <http://www-cscl95.indiana.edu/cscl95/fischer.html>.
- Fischer, G., Henninger, S. R., & Redmiles, D. F. (1991) "Cognitive Tools for Locating and Comprehending Software Objects for Reuse." In *Thirteenth International Conference on Software Engineering (Austin, TX)*, IEEE Computer Society Press, Los Alamitos, CA, pp. 318-328.
- Fischer, G., Nakakoji, K., & Ostwald, J. (1995) "Supporting the Evolution of Design Artifacts with Representations of Context and Intent." In G. M. Olson & S. Schuon (Eds.), *Proceedings of DIS'95 Symposium on Designing Interactive Systems: Processes, Practices, Methods, & Techniques*, ACM, New York, pp. 7-15. Available at: <http://www.cs.colorado.edu/~ostwald/papers/dis95-intent-abstract.html>.
- Fischer, G. & Nieper-Lemke, H. (1989) "HELGON: Extending the Retrieval by Reformulation Paradigm." In K. Bice & C. Lewis (Eds.), *Proceedings of CHI'89 Conference on Human Factors in Computing Systems*, ACM, New York, pp. 357-362.
- Fischer, G. & Reeves, B. N. (1995) "Creating Success Models of Cooperative Problem Solving." In R. M. Baecker, J. Grudin, W. A. S. Buxton, & S. Greenberg (Eds.), *Readings in Human-Computer Interaction: Toward the Year 2000*, Morgan-Kaufmann Publishers, Los Altos, CA, pp. 822-831.
- Furnas, G. W., Landauer, T. K., Gomez, L. M., & Dumais, S. T. (1987) "The Vocabulary Problem in Human-System Communication," *Communications of the ACM*, 30(11), pp. 964-971.
- Kintsch, W. (1998) *Comprehension: A Paradigm for Cognition*, Cambridge University Press, Cambridge, England.
- Nardi, B. A. (1993) *A Small Matter of Programming*, The MIT Press, Cambridge, MA.
- Ostwald, J. (2000) *Dynasites*, at <http://www.cs.colorado.edu/~ostwald/dynasites.html>.
- Repenning, A. & Sumner, T. (1995) "Agentsheets: A Medium for Creating Domain-Oriented Visual Programming Languages," *IEEE Computer, Special Issue on Visual Programming*, 28(3), pp. 17-25.
- Resnick, L. B., Levine, J. M., & Teasley, S. D. (Eds.) (1991) *Perspectives on Socially Shared Cognition*, American Psychological Association, Washington, D.C.
- Rittel, H. (1984) "Second-Generation Design Methods." In N. Cross (Ed.) *Developments in Design Methodology*, John Wiley & Sons, New York, pp. 317-327.
- Schön, D. A. (1983) *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, New York.
- Simon, H. A. (1996) *The Sciences of the Artificial*, (Third ed.), The MIT Press, Cambridge, MA.
- Snow, C. P. (1993) *The Two Cultures*, Cambridge University Press, Cambridge, UK.
- Wenger, E. (1998) *Communities of Practice — Learning, Meaning, and Identity*, Cambridge University Press, Cambridge, England.
- Williams, M. D. (1984) "What Makes RABBIT Run?" In *International Journal of Man-Machine Studies, Volume 21*, pp. 333-352.