DOMAIN-ORIENTED DESIGN ENVIRONMENTS: SUPPORTING INDIVIDUAL AND SOCIAL CREATIVITY

GERHARD FISCHER
Center for LifeLong Learning & Design (L^3D), Institute of Cognitive Science and Department of Computer Science
CB 430, University of Colorado, Boulder, CO 80309 USA
gerhard@cs.colorado.edu

Abstract. *Design is a ubiquitous activity. People can be creative in any sphere of life pursuing design activities. For the last decade, we have been engaged in the development of domain-oriented design environments to enhance individual creativity in a variety of different design domains.*

Based on the fact that the individual human mind is limited, extensions to our earlier approaches are presented, describing conceptual frameworks and innovative systems in support of social creativity. The conceptual framework is grounded in distributed cognition, organizational learning, and organizational memories. Derived from this conceptual framework, requirements for computational environments supporting social creativity are described, including the necessity for open systems; the seeding, evolutionary growth and reseeding model; and new conceptualizations of the World Wide Web. Specific system components and applications illustrate the implications of these assumptions and requirements. Experiences and assessments of these efforts describe the progress achieved over the last decade in our understanding of creativity and its computational support.

1. Introduction

Creativity is often associated with ideas and discoveries that are fundamentally novel with respect to the whole of human history (historical creativity). Here, however, we are concerned with creativity as it occurs in everyday work practice that is fundamentally novel with respect to the individual human mind or social community (psychological creativity) [Boden, 1991]. Analyzing outstanding creative people [Gardner, 1993] contributes toward establishing a framework for individual creativity, and understanding creativity in the context of social everyday activities is equally important for letting people become more productive and create better work products. The analysis of everyday design practices [Rogoff & Lave, 1984] has shown that knowledge workers and designers have to engage in creative activities to cope with the unforeseen complexities of everyday, real-world tasks.

Design activities [Simon, 1996] are ideally suited to pursue interesting issues in learning, working, collaboration, and creativity. Innovative computational environments provide a rich foundation to explore theoretical concepts, such as the creation of shared understanding [Arias et al., 1998; Resnick et al., 1991], conflict resolution [Arias, 1995], process (such as reflection-in-action [Schön, 1983]), and motivation [Csikszentmihalyi, 1990]. The importance of understanding and developing theories of design [Cross, 1984; Greenbaum & Kyng, 1991] for creativity is twofold: (1) they can be used in the creation of innovative computational environments, and (2) they support essential learning mechanisms for design activities, such as learning on demand [Fischer, 1991] and collaborative working and learning [Brown & Duguid, 1991].

In this paper, I will first characterize individual creativity and our earlier work to support it. Social creativity is introduced to address the limitations of individual creativity. Some system requirements for social creativity are formulated and some of our system-building efforts in response to these requirements are described. An assessment of our efforts is discussed and our progress over the last decade (since the first creativity meeting on Heron Island in 1989) is analyzed.

2. Individual Creativity

2.1 EMPOWERING INDIVIDUALS

Illich [Illich, 1973] defined convivial tools as those “which give each person who uses them the greatest opportunity to enrich the environment with the fruits of his or her vision.” Because people and tasks are different, especially in design (e.g., Schön [Schön, 1983] describes design problems as “a universe of one”), these visions are unique and idiosyncratic. People become engaged and excited about personally meaningful ideas, they strive for self-expression, and they want to work and learn in a self-directed way [Fischer & Nakakoji, 1992]. Rather than settling on the “lowest common denominator” approach and trying to make everyone happy, computational media have the potential to reindividualize and empower people [Norman, 1993].

Empowering individuals requires that we redefine the roles of “high-tech scribes.” Conceptual frameworks and computational environments are needed that will give domain designers more independence from computer specialists. Just as the pen was taken out of the hands of the scribes in the Middle Ages, the power of the high-tech computer scribes should be redefined. To turn computers into convivial tools requires that people can use, change, and enhance their tools and build new ones without having to become professional-level programmers. Convivial tools will contribute to the creation of a climate in which a much broader range of people will wish to become “scribes” so that they can engage in creative activities by taking advantage of computational media.

2.2 DOMAIN-ORIENTED DESIGN ENVIRONMENTS

In our past research efforts in the L^{3}D Center at the University of Colorado at Boulder we have developed conceptual frameworks to empower individuals by developing domain-oriented design environments (DODEs) [Fischer, 1994a] in a variety of different domains. By being domain-oriented, they support *human problem-domain communication*, making the computer invisible and bringing tasks to the forefront. DODEs (created over time as a joint effort among clients, domain designers, and environment developers) empower individuals [Fischer & Nakakoji, 1992] by

1. letting them articulate a partial description of their tasks with the help of a specification component,
2. supporting the creation of an artifact with a construction component,
3. signaling potential breakdowns with a critiquing component,
4. supporting the exploration of argumentation and design rationale,
5. providing additional feedback with a simulation component, and
6. reminding and guide them through the process with a checklist.

2.3 EXPLOITING BREAKDOWNS WITH CRITICS AND LEARNING ON DEMAND

A system supporting creativity will be sufficiently open-ended and complex that users will encounter *breakdowns* [Fischer, 1994b; Popper, 1965]. The system must provide means for allowing users to understand, extricate themselves, and learn from breakdowns. Rather than attempting to eliminate trouble, the system should help users manage troubles and exploit breakdowns as opportunities rather than failures.

As any professional designer knows, breakdowns—although at times costly and painful—offer unique opportunities for reflection and learning

[Petroski, 1985]. Breakdowns can occur at many levels: at the tool level (e.g., a computational environment does not provide the functionality needed), and at the domain level (e.g., domain designers are lacking or overlooking some important domain knowledge). To exploit these opportunities, we have developed conceptual frameworks and innovative systems to support critiquing and learning on demand.

Critiquing systems [Fischer et al., 1993] offer advice and information by supporting reflection-in-action [Schön, 1983], thereby allowing users to explore the contextualized argumentation and design rationale associated with their actions. Using design environments, designers create artifacts serving as externalizations of their thoughts [Bruner, 1996]. These artifacts can be critiqued by computational critics, increasing the "back-talk" of the design situation [Schön, 1983]. Only designers, not consumers, can engage in learning by doing, which has emerged as one of the most effective learning strategy.

Learning on demand [Fischer, 1991] allows designers to explore contextualized information that is directly relevant to their breakdown situations. Information overload, high functionality applications, and rapid technological changes have created new problems and challenges for learners of all ages. New learning environments 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). Learning on demand is the only viable strategy in a world where we cannot learn everything. It is a promising approach to supporting creativity for the following reasons: (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, thereby leading to better decision making, better products, and better performance.

3. Social Creativity

The basic foundation for social creativity is that people think, work, and learn in conjunction or partnership with others and with the help of culturally provided tools and artifacts. Based on the argument that the individual human mind is limited, we have to put knowledge into the world (in the form of externalizations, oeuvres, and sharable artifacts [Bruner, 1996]) and support distributed cognition [Norman, 1993] where the distribution occurs between humans and between humans and artifacts.

This raises challenges in finding ways to tailor the external information repositories to the knowledge in the heads of specific individuals and to the demands of tasks requiring representations that fit these needs, can be adapted to these needs, or adapt themselves to these needs based on models of the task and the user.

3.1 THE INDIVIDUAL HUMAN MIND IS LIMITED

The power of the unaided mind is highly overrated—without external aids, memory, thought, and reasoning are all constrained [Norman, 1993]. The Renaissance scholar does not exist anymore. Human beings have a bounded rationality—making satisfying instead of optimizing a necessity [Simon, 1996]. 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, collaboration will become a necessity, and practitioners will make increasing use of reference aids, such as printed and computational media supporting external cognition.

Design is one such domain. Complexity in design arises from the need to synthesize different perspectives of 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. To cope with the complexity, designers use high-functionality applications to create external representations. The broad requirements for knowledge workers (e.g., a multimedia designer [Nakakoji et al., 1996] who must deal with text, color, sound, film, animations, etc.) require that they use a number of high functionally applications, and (as argued above) individuals have difficulty in learning and using just *one* of these systems. To express and externalize our evolving designs, domain-oriented design environments need to be developed, used, and assessed to support social creativity. These extended domain-oriented design environments should allow us to (1) address the “symmetry of ignorance” [Rittel, 1984], (2) create shared understanding [Resnick et al., 1991], (3) analyze artifacts [Fischer, 1994b], and (4) incrementally construct domains and domain models that do not a priori exist but instead are socially constructed over time by communities of practice [Lave, 1991]. To account for these requirements, our approach emphasizes the prominent role that domain practitioners must play in constructing an initial model of the domain, rooted in work practices, and evolving this model over time to suit their changing needs.

Communities of Practice. Until recently, computational environments focused on the needs of individual users. As computers were used for more complex tasks by more people, it became apparent that environments supporting social communities such as communities of practice, groups, and organizations were needed (as demonstrated by research efforts in computer-supported cooperative work, institutional memories, interoperability, and so forth). However, this perspective does not necessitate the development of environments in which the interests of the group inevitably supersede those of the individual. Individuality makes a difference, and organizations get their strength to a large extent from the creativity and engagement of their individuals. We need to understand the “mutually constitutive relationship of the individual and the social” [Brown & Duguid, 1992].

The creation, integration, and dissemination of knowledge is more important than ever to organizations, and yet the traditional ways of managing knowledge are proving inadequate to meet the needs. In the past, organizational knowledge was considered to be generated at the top of organizations, housed in documents containing canonical job and task descriptions, and delivered to workers in occasional classroom training sessions. Now, organizations are finding that the knowledge that workers need to perform their jobs cannot exist in any canonical form. Due to the rapid pace of change in the world, workers often face problems that their managers and training experts do not, and could not, anticipate. The knowledge that workers need must instead be accessed if it exists, or created if it doesn't, by workers themselves as they grapple with unforeseen problems. This knowledge is practical and is the result of innovation, creativity, and communication among co-workers.

Lack of Support for Collaborative Design. In designing artifacts, designers rely on the expertise of others [Galegher et al., 1990; Resnick et al., 1991] by referring to textbooks, standards, legal constraints, and especially previous design efforts. Project complexity forces large and heterogeneous groups to work together on projects over long periods of time. Knowledge bases should include not only knowledge about the design process but also knowledge about artifacts of that process—parts used in designing artifacts, subassemblies previously created by other design efforts, and rationale for previous design decisions [Fischer et al., 1992]. Designers generally have a limited awareness and understanding of how the work of other designers within the project—or in similar projects—is relevant to their own part of the design task. The large and growing discrepancy between the amount of such relevant knowledge and the amount any one designer can possibly remember imposes a limit on progress in design. Overcoming this limit is a central challenge for developers of systems that support collaborative creativity [Nakakoji, 1998].

3.2 ORGANIZATIONAL LEARNING AND ORGANIZATIONAL MEMORIES

Organizational learning focuses on recording knowledge gained through experience (in the short term), and actively making that knowledge available to others when it is relevant to their particular task (in the long term) [Fischer et al., 1996b]. A central component of organizational learning is a repository for storing knowledge_ an organizational memory. However, the mere presence of an organizational memory system does not ensure that an organization will learn. Today, information is not a scarce commodity; the problem is not just to accumulate information, but to deliver the *right* knowledge at the right time to the right person in the right way. Organizational learning happens only when the contents of organizational memory are utilized effectively in the service of doing work. Efficient support for organizational learning raises many unresolved issues: how can we create a working and learning culture, in which individuals are willing and encouraged to share; how do we effectively collect individual knowledge and make it easily accessible to the entire organization?

For sustained organizational learning, three seemingly disparate goals must be served simultaneously. Organizational memory must (1) be extended and updated as it is used to support work practices; (2) be continually reorganized to integrate new information and new concerns; and (3) serve work by making stored information relevant to the new task at hand. Organizational learning is a continuous cycle in which organizational memories play a pivotal role:

- Individual projects serve organizational memory by adding new knowledge that is produced in the course of doing work, such as artifacts, practices, rationale, and communication.
- Organizational memories are sustained in a useful condition through a combination of computational processes providing information and people actively contributing.
- Organizational memory serves work by providing relevant knowledge when it is needed, such as solutions to similar problems, design principles, or advice.

The intimate relationship between organizational memory and work practices implies that the contents of organizational memory must be easily accessible within the context of work. Computational support for organizational learning, therefore, must tightly integrate tools for doing work with tools for accessing the contents of organizational memory.

A principal challenge for organizational learning is to capture a significant portion of the knowledge generated by work done within a community. Experience with organizational memories and collaborative work has exposed two barriers to capturing information:

- Individuals must perceive a large enough direct benefit in contributing to organizational memory to outweigh the effort [Grudin, 1994].
- The effort required to contribute to organizational memory must be minimal so it will not interfere with getting the real work done [Carroll & Rosson, 1987].

The consequence of these barriers means that processes of information capturing, structuring, and delivery must be computationally supported as much as possible or they simply will not get done.

Organizational memories are information systems that are used to record knowledge for the purpose of making this knowledge useful to individuals and projects throughout the community of practice and into the future. Ideally, an organizational memory allows individuals within the community to benefit from the experiences and insights of others, by *actively* informing work practices at the point when the information is actually needed [Fischer & Nakakoji, 1994]. That is, an organizational memory should not be simply a passive repository of information, but an interactive medium within which collaborative work can actually be conducted and through which communication about the work can take place and be situated. Systems that support *organizational learning* [Senge, 1990] and *organizational memories* [Terveen et al., 1995] will be useful for professionals working on complex tasks in large team environments. An example of an organizational memory is GIMMe, the Group Interactive Memory Manager [Fischer et al., 1996b; Lindstaedt, 1998], which captures group email, automatically categorizes it, and then provides context-sensitive search capabilities. These systems will have to be enhanced to capture richer types of information and provide more powerful categorization and search techniques.

3.3 FROM CONSUMERS TO DESIGNERS

Social creativity is impossible in communities in which most of their members regard themselves as consumers. Consumers must evolve into power-users [Nardi, 1993] and co-developers [Henderson & Kyng, 1991] who use artifacts and at the same time modify and extend them. A strict separation between these two groups is undesirable and unproductive. One of the biggest potentials of information technology is giving people the option to become designers by changing and enhancing a software system. Software, after all, as already indicated in its name, should be "soft." One of the major contributions that information technology can lend to the world is to deeply understand and exploit the potential of the malleable nature of software.

Individuals acting as designers must acquire a *new mindset*—they are no longer passive receivers of knowledge, but instead are active researchers, constructors, and communicators of knowledge. Knowledge is no longer handed down from above, but instead is constructed collaboratively in the contexts of work. Empowering individuals with convivial tools is grounded in the fundamental belief that humans (albeit not all of them, not at all times, not in all contexts) want to be and act as designers [Fischer, 1998].

By arguing for the desirability for humans to be designers, it should be stated explicitly that there is nothing wrong with being a consumer and that we can learn and enjoy many things in a consumer role (e.g., listening to a lecture, watching a tennis match, or attending a concert). It is also a mistake to assume that being a consumer or being a designer would be a *binary choice*; it is rather a continuum ranging from passive consumer, to active consumer, to end-user, to user, to power users, to domain designer, to medium designer, all the way to meta-designer. Problems occur (1) when someone wants to be a designer but is forced to be a consumer, and (2) when being a consumer becomes a universal habit and mindset dominating a human life completely. For example, in thinking about the future (including the role of new media and new technologies for our future societies), the consumer asks: "Is a new future coming?" whereas the designer asks: "How can we invent and create a new future?" The designer understands that the future is not out there to be "discovered"—but it has to be *invented and designed*. Accepting this will raise the issue: *who* will design the future?

4. System Requirements for Social Creativity

Social creativity rests on the premise that one of the major roles for computational media is not to deliver predigested information to individuals, but to provide the opportunity and resources for design activities embedded in social debates and discussions in which all people can act as designers rather than being confined to consumer roles.

Unfortunately, many users perceive computer systems as unfriendly, and uncooperative, and their use as too time consuming; they spend more time fighting the computer than solving their problems. Many users depend on specialists ("high-tech scribes") for help, and the world of computing is separated into a population of elite scribes who can act as designers and a much larger population of intellectually disenfranchised computerphobes who are *forced* into a consumer role.

One of the biggest challenges facing systems in support of social creativity is to allow end-users to become co-developers of systems. Figure 1 differentiates between two stages in the design and use of an artifact. At *design time*, system developers create environments and tools including help systems, guided tours, forms, and so on, and they have to make decisions for users (who may want to be consumers or designers), for situational contexts and for tasks that they can only anticipate. For print media, a fixed context is decided at design time, whereas for computational media, the behavior of a system at *use time* can take advantage of contextual factors (such as the background knowledge of a user, the specific goals and objectives of a user, the work context) *only known at use time*. The fundamental difference is that computational media have interpretive power: they can analyze and critique the artifacts created by users [Fischer et al., 1993]—and users acting as designers will create artifacts off all kinds. The challenge is to build new innovative systems that allow the users to articulate contextual factors (e.g., in using a specification component [Nakakoji, 1993] and/or infer this information from the environment), which will serve as objects for interpretation.

A Motivating Example: 700 Help Desk People. One of our collaborating companies employs about 700 help desk people. These employees help customers to solve their problems. In the example, help desk person N expends considerable effort to solve a difficult customer problem. How should this creative act be documented and shared with the

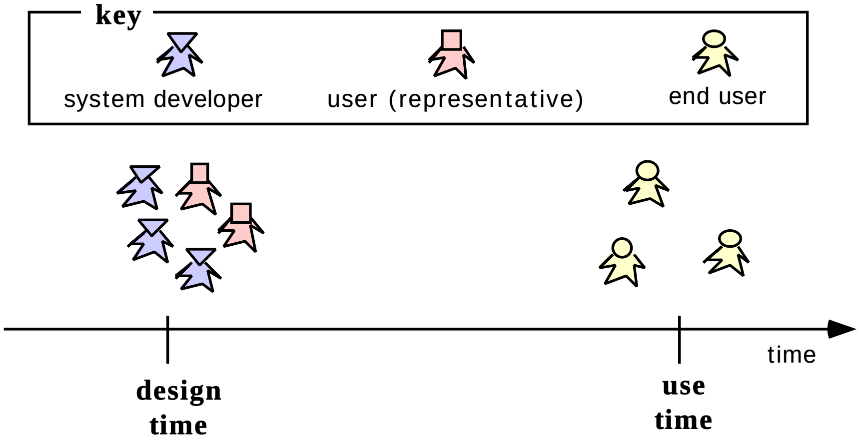


Figure 1: Design and Use Time

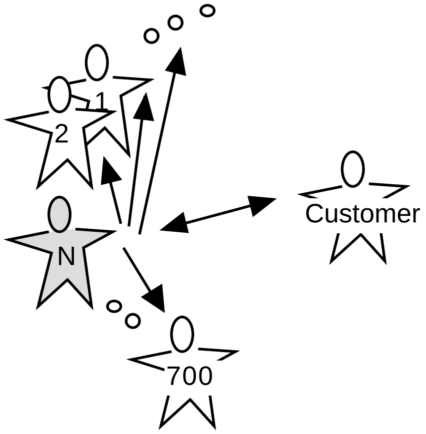


Figure 2: Broadcast: Information Overflow of Decontextualized Information

other help desk people? Should person N *broadcast* this problem and its solution to the 699 other help desk people, as illustrated by Figure 2?

We claim the answer is no, because in general this information will not be relevant to the other help desk people. The more promising strategy is captured in Figure 3: the information should be captured in an information repository (such as an organizational memory) and it should be made available (either upon request or volunteered by the system) when other help desk people encounter a problem in the future to which the solution created and documented by person N is relevant.

These two examples are illustrative of the kinds of experiences that are important to members of organizations. The core challenge for social creativity is to collect creative solutions by individuals and make them available to colleagues who encounter similar problems. This core challenge raises difficult technical issues such as:

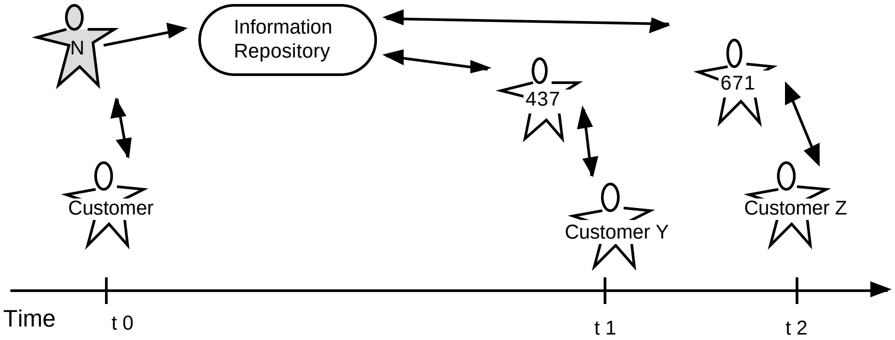


Figure 3: The Challenge: Supporting Learning on Demand

1. computationally tractable representations of experience [Shipman, 1993];
2. retrieval technologies that notice deep as well as surface similarities [Landauer & Dumais, 1997];
3. capturing a significant portion of the knowledge that practitioners generate in their work [Fischer et al., 1992];
4. developing a culture in which individuals are motivated to work for the good of the group or organization [Grudin, 1994].

4.1 CLOSED VERSUS OPEN SYSTEMS

Closed systems in which the essential functionality is anticipated and designed at design time (see Figure 1) are inadequate in coping with the tacit nature of knowledge and the situatedness of real-world problems that can promote and enhance social creativity. Computational environments need to be open systems because they model parts of our world. Our world evolves in numerous dimensions as new artifacts appear, new knowledge is discovered, and new ways of doing business are developed. Successful software systems need to evolve. Enhancing and evolving systems need to become "first class design activities," extending system functionality in response to the needs of its users. This will require support for end-user modification and programming, which can best be achieved by putting owners of problems in charge.

In our research we have carefully analyzed why simulation environments such as SimCity will not be used for real planning and working environments. SimCity acknowledges that users might want to design their own structures by providing the SimCity Urban Renewal Kit (an add-on module to increase user control), but it does not allow users to define new behavior. For example: if users notice that the crime rate is too high, they can build more police stations to fight crime, but they cannot increase social services or improve education to prevent crime. These empirical studies have led us to the following claims: (1) *software systems must evolve; they cannot be completely designed prior to use* (software users and designers will not fully determine a system's desired functionality until that system is put to use); (2) *software systems must evolve at the hands of the users* (end users experience a system's deficiencies; subsequently, they have to play an important role in driving its evolution); and (3) *software systems must be designed for evolution* (through our research, we have discovered that systems need to be designed *a priori* for evolution).

4.2 SEEDING, EVOLUTIONARY GROWTH, AND RESEEDING

Design environments themselves grow over time by their use through extensions made by individuals. This raises the interesting question of which of these individual extensions should be embedded in the generic environment to be accessible by all people using it. There is a price to pay for individualization: environments cannot be shared as easily, and it may require substantial work to establish a common background and a shared understanding.

Because design in real-world situations deals with complex, unique, uncertain, conflicted, and unstable situations of practice, DODEs will never be complete because design knowledge is tacit (i.e., competent practitioners know more than they can say) [Polanyi, 1966], and additional knowledge is triggered and activated by actual use situations leading to breakdowns [Fischer, 1994b]. Because these breakdowns are experienced by the users and not by the developers, computational mechanisms that support end-user modifiability are required as an intrinsic part of a DODE.

The <u>s</u>eeding, <u>e</u>volutionary growth, and <u>r</u>eseeding (SER) model (see Figure 4) is a process model developed to support evolvable systems. Three intertwined levels can be distinguished whose interactions form the essence of the SER model:

1. On the *conceptual framework level*, the multifaceted, domain-independent architecture constitutes a framework for building evolvable complex software systems.
2. When this architecture is instantiated in a domain (e.g., computer network design, multimedia design), a domain-oriented design environment (representing an application family) is created on the *domain level*.
3. Individual artifacts in the domain are developed by exploiting the information contained in the generic DODE.

Figure 4 illustrates the interplay of these three levels. Darker grey indicates knowledge domains close to the computer, whereas white emphasizes closeness to the design work in a domain. The figure illustrates the role of different professional groups in the evolutionary design: the *environment developer* (close to the computer) provides the domain-independent framework and instantiates it into a DODE in collaboration with the help of the domain designers (knowledgeable domain workers who use the environment to design artifacts). Domain designers are the “end users” of a design environment. The artifact is eventually delivered to the client.

Breakdowns occur when domain designers cannot carry out the design work with the existing DODE. Extensions and critics drive the evolution on all three levels: Domain designers directly modify the artifacts when they build them (artifact evolution), they feed their modifications back into the environment (domain evolution), and—during a reseeding phase—even the architecture may be revised (conceptual framework evolution). In Figure 4, the little building blocks represent knowledge and domain elements in any of the components of the multifaceted architecture.

The evolution of complex systems in the context of this process model (more detail can be found in [Fischer et al., 1994]) can be characterized as follows:

Seeding. A seed will be created through a participatory design process between environment developers and domain designers. It will evolve in response to its use in new network design projects because requirements fluctuate, change is ubiquitous, and design knowledge is tacit. Postulating the objective of a seed (rather then a complete domain model or a complete knowledge base) sets our approach apart from other approaches in software engineering and artificial intelligence and emphasizes evolution as the central design concept.

Evolutionary growth. *Evolutionary growth* takes place as domain designers use the seeded environment to undertake specific projects for clients. During these design efforts, new requirements may surface, new components may come into existence, and additional design knowledge not contained in the seed may be articulated. During the evolutionary growth phase, the environment developers are not present, thus making end-user modification a necessity rather than a luxury (see Figure 5).

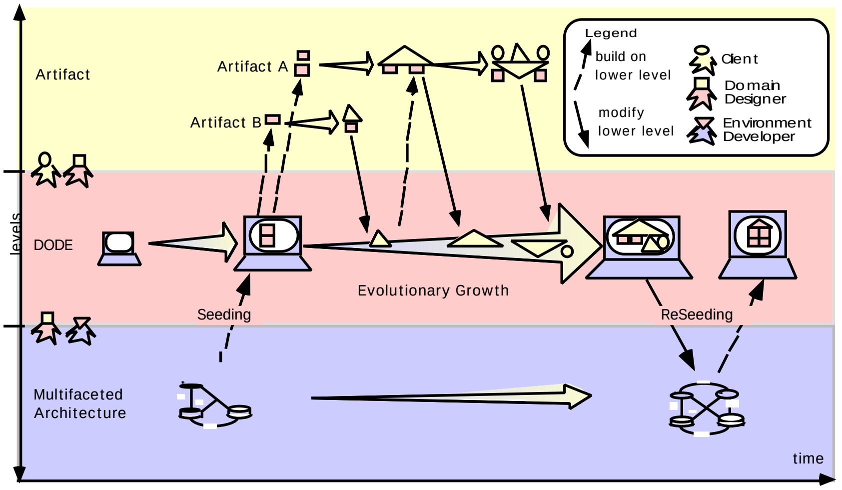


Figure 4: The SER Model: A process model for the development and evolution of DODEs

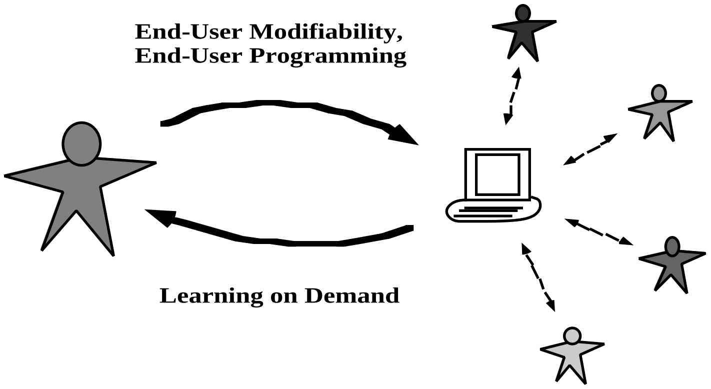


Figure 5: Duality between Learning and Contributing

Visual AgenTalk (VAT) (see Figure 8) addresses this problem, building on our experience with end-user modifiability and end-user programming [Eisenberg & Fischer, 1994].

Figure 5 characterizes the duality and the distributed nature of knowledge: a specific user can learn (specifically learn in context and on demand) from a computational environment (which contains knowledge and tools contributed by many members of the community of practice), but if this user considers her/himself a designer, she/he will also contribute to the environment (assuming mechanisms are available that allow her/him to do so with a reasonable effort). This perspective illustrates the concepts and need for co-adaptive systems: (1) users learn from the systems; (2) users acting as innovators, co-developers and designers adapt and evolve the systems; and (3) support for organizational learning allows users to share these adaptations with others.

Reseeding. In a deliberate effort to revise and coordinate information and functionality, the environment developers are brought back in to collaborate with domain designers to organize, formalize, and generalize knowledge added during the evolutionary growth phases. Organizational concerns [Grudin, 1991] play a crucial role in this phase. For example, decisions have to be made as to which of the extensions created in the context of specific design projects should be incorporated in future versions of the generic design environment. Drastic and large-scale evolutionary changes occur during this reseeding phase.

4.3 NEW CONCEPTUALIZATIONS OF THE WORLD WIDE WEB

Technologies that enhance the World Wide Web as a medium that supports learning [Pea, 1996] include (1) *object repositories* to assist the creation and sharing of educational resources on the Web, and (2) *tools* *for intelligently locating learning resources* that support individual learners based on their background knowledge. Existing object repositories include the Agentsheets Behavior Exchange [Repenning & Ambach, 1997] (see Figure 9 and Figure 10); a Web site that allows Agentsheets developers to exchange entire simulations and individual simulation components; and the Educational Object Economy, a shared repository of interactive, educational applications supported by Apple Computer, Inc. and a consortium of companies and universities including CU-Boulder. These object repositories will have to be enhanced in order to support not only the discovery of relevant objects, but also the reuse and modification of these objects. Some of the tools that assist individual learners by providing the discovery of relevant learning materials will be based upon latent semantic analysis [Landauer & Dumais, 1997] to carry out conceptual, content-based retrieval**.**

The Web in its current form does not support evolutionary design. Traditional Web-based use engages the Web as a broadcast medium in which content is predetermined at design time (see Figure 1) and placed on static Web pages. Most popular general-purpose Web tools provide support for the easy generation of this static content. As a broadcast medium, the Web serves as a distribution channel and provides few opportunities for designers to interact with the information because the content was not originally designed to be interactive. Responding to the need for feedback from consumers, many Web sites are evolving into forms that augment content with some communication channels. *Broadcast with feedback* provides links from consumer to producer such as allowing learners to provide feedback and ask questions by filling out forms. Although users can react to information provided by the author, this presentation model provides little support for evolution.

To support social creativity, users need to be able to use the Web to collaborate on projects by *actively* contributing and by learning from all contributors. The evolution of content and ideas is now the responsibility of the participating community of practice, focusing on the distributed generation of content and the reflection upon it. Such an environment is needed to support the SER model. When a wide variety of individuals collaborate in a cooperative forum, the unique skills of the members all become valuable resources in making the Web resources useful in the current context. This model of the Web poses a number of technical challenges, including the ability (1) to add to an information space without going through an intermediary, (2) to modify the structure of the information space, and (3) to modify at least some of the existing information.

4.4 DECENTRALIZED CONSTRUCTED INFORMATION REPOSITORIES

The SER model is a useful framework for understanding the processes inherent in the development of *open systems* [Fischer & Scharff, 1998]. For example, the development of open-source software systems such as the Linux operation system [Raymond, 1998] provides interesting proof that reliable, useful, and usable complex systems can be built in a decentralized “Bazaar style” by many rather than in a centralized, “Cathedral style” by a few. The Linux development model treats users simultaneously as co-developers and self-directed learners (see Figure 5) and is currently being tested in a number of new areas (e.g., in the *Netscape Communicator*; for more information see http://www.mozilla.org/).

Examples of first steps toward systems illustrating the power of social creativity based on community participation are beginning to take form in the software design community. Java developers use the Web to facilitate the learning of Java and to share the components created in Java. Due to the contributions of developers around the world, the Java programming community has used community repositories of knowledge to produce technical advances in a very short period of time. **Gamelan** (http://www.gamelan.com) is one of the first community repositories of Java-related information. The primary users of Gamelan are Java developers looking for information about what other people are doing with Java. Gamelan is a forum to facilitate the self-directed learning of members of the emerging Java community. The software developers who use the content are also the primary contributors, continuously adding new resources to the Gamelan repository. Gamelan was originally designed to be the official clearinghouse for all third-party uses of Java, and the site attempts to support any work that uses Java.

The **Educational Object Economy** (EOE) (http://www.eoe.org/) provides a more focused system than Gamelan. Currently realized as a collection of Java objects (mostly completed applets) designed specifically for education, the target users of the EOE are teachers (presumably acting as consumers of completed applets) wishing to use new interactive technology and instructional designers interested in producing educational software. The EOE's primary goal is to provide educators with a collection of useful resources ready to be used to help students learn.

The Agentsheets **Behavior Exchange** (http://agentsheets.cs.colorado.edu/agents) is an initial prototype of a domain-specific system for sharing computational artifacts and is discussed further in Section 5.3.

One important common feature of all three of these systems is their support for evolution. As new knowledge becomes available, members of the community may share new developments with each other. In all three systems, the repository administrators set up an initial *seed* that structures how information is added, presented, and searched by users. The goal is to create useful information repositories in a decentralized fashion [Resnick, 1994]. All three systems allow *evolution* by the community who uses the information, although the centralized authority plays different roles in the different systems. In Gamelan (and to a lesser extent, the EOE), all new resources are verified and filtered by the repository administrator. In the Behavior Exchange, contributions are unrestricted. In fact, contributors have some control of the categorization itself because of the ability to add new projects or categories. Evolution in all the repositories is limited to the *addition* of content; users can add new resources. The ability to *refine* content is limited in all the systems. Because users can add only new resources, and all resources stand on their own, it is impossible to track the changes of individual components. Finally, over the past few years, all three systems have gone through dramatic redesign or *reseeding* phases in which content is checked and reformulated and revised, entries are related to each other (which might possibly have been captured during the evolution phase), categorizations scheme change, information presentation goes through major changes, and different searching methods are employed. In all three cases, reseeding has been performed by the environment developers based on feedback from the community.

Because all three systems are envisioned as tools that evolve at the hands of a community of users, all three are prime candidates to study the challenges, strengths, and weaknesses of open systems. The SER process model that accounts for the evolution of DODEs is a useful framework for understanding the changes that have taken place in these systems in support of social creativity.

5. Computational Environments Supporting Social Creativity

5.1 ENVISIONMENT AND DISCOVERY COLLABORATORY

The Envisionment and Discovery Collaboratory (EDC) [Arias et al., 1998] is an attempt to support social creativity by creating shared understanding among various stakeholders, contextualizing information to the task at hand, and creating objects-to-think-with in collaborative design activities. The EDC framework is applicable to different domains, but our initial effort has focused on the domains of urban planning and decision making, specifically in transportation planning and community

Figure 6: The EDC Environment

development. Creating shared understanding requires a culture in which stakeholders see themselves as reflective practitioners rather than all-knowing experts [Schön, 1983]. Collaborative design taking place in such a culture can be characterized by an "asymmetry of knowledge" or a "symmetry of ignorance" [Rittel, 1984]: stakeholders are aware that while they each possess relevant knowledge, none of them has all the relevant knowledge.

Figure 6 shows the current realization of the EDC environment. Individuals using the EDC convene around a computationally enhanced table, shown in the center of the figure. This table serves as the Action Space for the EDC. Currently realized as a touch sensitive surface the Action Space allows users to manipulate the computational simulation projected on the surface by interacting with the physical objects placed on the table. The table is flanked by a second computer driving another touch-sensitive surface shown in the upper right of Figure 6. This computational whiteboard serves as the EDC's Reflection Space. In the figure, neighbors are filling out a Web-based transportation survey that is associated with the model being constructed. The Reflection and Action spaces are connected by communication between the two computers using the Web as a medium. The entire physical space, through the immersion of people *within* the representations of the problem-solving task, creates an integrated human/computer system grounded in the physical world.

As argued before, much development of technology for learning and design builds on or is constrained by the "single user/single computer" interaction model. The Envisionment and Discovery Laboratory emphasizes the creation of shared interaction, social structures, and cultural embedding for learning within the context of communities of learners. It is being developed as a learning and design support medium where 3-D physical objects interact dynamically with virtual ones over an integrated sensory/display worksurface as the computational gameboard. Based on 10 years of experience in building physical simulation games, we see that powerful collaborative learning and shared decision making can be supported by shared interaction and integration with computational models. Together these form a collaborative environment that builds on both distributed and face-to-face collaborations in classrooms or public sites (e.g., libraries).

Crucial processes relevant for social creativity and supported by the EDC are:

- dealing with a set of possible worlds effectively (i.e., exploring design alternatives) to account for the design is an argumentative process, where we do not prove a point but we create an environment for a design dialog [Simon, 1996];
- using the symmetry of ignorance (i.e., that all involved stakeholders can contribute actively) as a source of power for mutual learning by providing all stakeholders with means to express their ideas and their concerns [Rittel, 1984];
- incorporating an emerging design in a set of external memory structures, and recording the design process and the design rationale [Fischer et al., 1996a];
- creating low-cost modifiable models, which help us to create shared understanding, have a conversation with the materials [Schön, 1983], and replace anticipation (of the consequences of our assumptions) by analysis;
- using the domain-orientation to bring tasks to the forefront and supporting human problem-domain communication [Fischer & Lemke, 1988];
- increasing the "back-talk" of the artifacts with critics [Fischer et al., 1993];
- using simulations to engage in "what-if" games [Repenning, 1994].

The EDC is a contribution to create a new generation of collaborative DODEs. It shifts the emphasis away from the computer screen as the focal point and creates an *immersive* environment in which stakeholders can incrementally create a shared understanding through collaborative design. It is an environment that is not restricted to the delivery of predigested information to individuals, but it provides opportunities and resources for design activities embedded in social debates and discussions in which *all* stakeholders can actively contribute rather than being confined to passive consumer roles.

5.2 DYNASITES

The reflection space in EDC is supported by DynaSites [Ostwald, 1997], extensible, Web-based information spaces in which users can add to information spaces, create new structures, and link information with the Web. DynaSites investigate computational support for collaborative working, learning, and knowledge construction by postulating that information spaces supporting these activities should grow and be shaped over time by the people who use them. DynaSites information spaces are Web sites (they can be viewed through a Web browser), but they differ from most Web sites because they are dynamic and evolvable by users. DynaSites information spaces are dynamic because their pages are built at *use time* (from a database) whereas typical Web sites are static—their links and displays are determined at *design time* (see Figure 1).

The original aim of DynaSites was simply to overcome the static nature of Web pages. The first of the DynaSites was DynaGloss, an extensible glossary of terms (see Figure 7), where users could define new terms, annotate, or redefine existing definitions. Subsequent DynaSites were based on “discussion forums,” where discussion topics could be raised and then commented on. A technique for integrating the discussion forums with the glossary was developed that automatically identifies glossary terms occurring in the contents of the forums, and creates a link to the definition in the glossary. Experience using DynaSites for other types of information spaces (e.g., the construction and presentation of class project reports) showed that the read-only nature of discussion forums is inappropriate for a community of practice that learns and contributes on an ongoing basis.

DynaSites support communities of practice to partition their information into different documents and to create semantic relations (via hypertext links) between the documents. For example, different communities can maintain their own discussion forums, but share information by linking the forums together. Or a single community can create different forums for specific artifacts (e.g., networks) but link information to a forum containing general design principles. In this way, DynaSites address the following issues relevant to social creativity:

Organizational Memory

Current Definition

A shared information space that supports a group of people (an organization) to do work. The information space should be "living" in the sense that it is an evolving product of the work done by the members of the organization as opposed to simply being a static storage of information.

Known Uses of *Organizational Memory* in DynaSites

A question of terminology (in DynaClass)
Group Decision Support Systems (in Sources)
Commercial product for Organizational Memory (in DynaClass)
1/26/98 Meeting Minutes
Living Design Memory: Framework, Implementation, Lessons Learned (in Sources)
Work Promised
General Goals

Figure 7: DynaGloss—A Dynamic, Collaboratively Evolvable Glossary

Information Types: Homogeneous information spaces force users to employ the same formalisms to express many different types of knowledge (such as people, design principles, discussions, rationale). Although this may lower the costs of authoring, it also lowers the amount of support the system can provide in working with information.

Computation: Like newsgroups, DynaSites are based on the database model, but add "use-time" processing of information content. Use-time processing enables DynaSites displays to be automatically updated in accordance with a variety of events, including the changing state of other information in the system. An example of how use-time computation of information contents is used by DynaSites is the glossary mechanism: when a term is added to the glossary, occurrences of that term in other documents (such as a discussion) are automatically turned into links.

Support for Information Space Evolution: Information space evolution is more than simply an accumulation of more and more information. Evolutionary processes should include making new associations between pieces of information, creating new structures that group related information together so that similarities and generalizations become apparent, and removing obsolete or erroneous information from the information space. These processes allow the quality of information to improve over time, and to make important information more accessible to users.

End-user Modification: An important requirement for information space evolution is that users (domain designers) are able to modify their information spaces in a variety of ways beyond simply adding information to an existing structure. Whereas this is impossible for most Web sites, it is an essential design principle of DynaSites.

5.3 AGENTSHEETS, VISUAL AGENTALK AND BEHAVIOR EXCHANGE

The action space of the EDC is built using Agentsheets [Repenning, 1997], a software environment for creating simulations, games, and DODEs. Agentsheets applications include a collection of autonomous computational processes, called agents, that are comprised of a look, their on-screen representation, and a programmed behavior. Agents in Agentsheets are programmed in *Visual AgenTalk*, a programming environment suitable for end-user programmers.

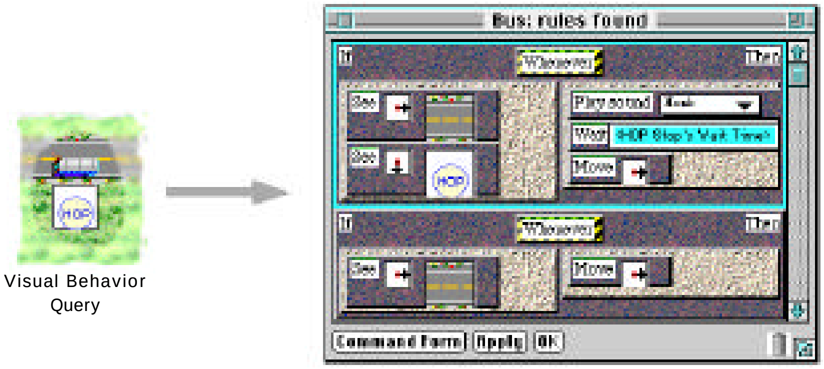
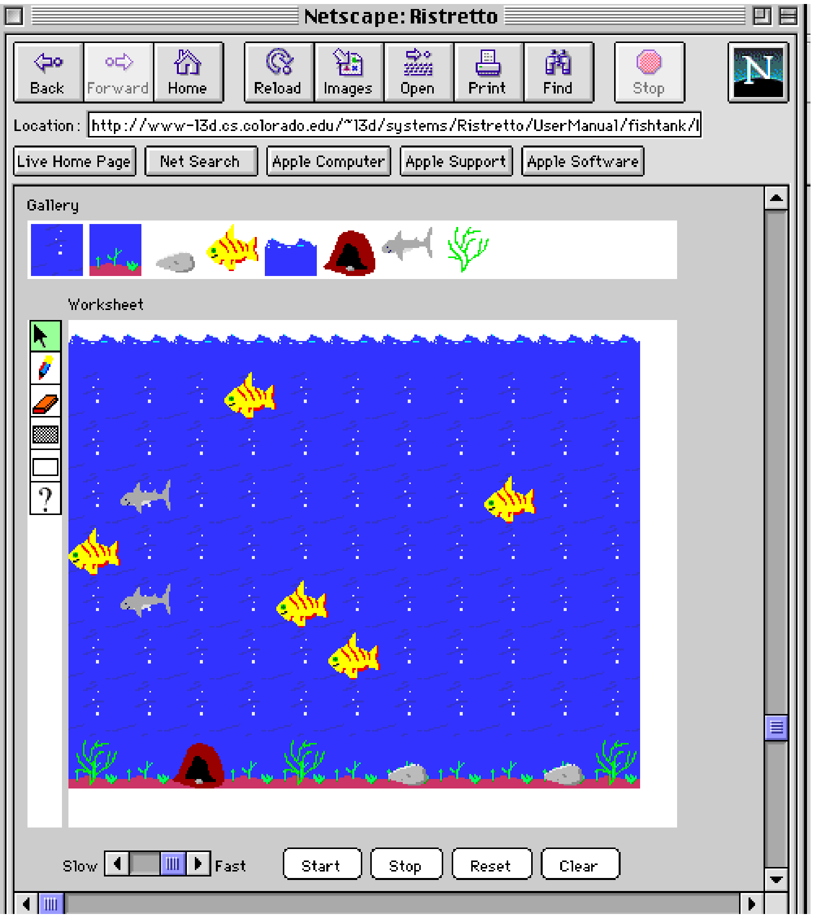


Figure 8: Visual AgenTalk

Whereas DynaSites is a substrate for creating textual, evolvable web-based information spaces, the Agentsheets *Behavior Exchange* extends these evolvable spaces to include not only textual artifacts but computational artifacts, including simulations and agents. Agents in Agentsheets are programmed in Visual AgenTalk (see Figure 8, which shows some rules used in the transportation planning domain of the Envisionment and Discovery Collaboratory) suitable for end-users, reducing the demands on end-users to migrate from consumers to designers.



Location: http://www-l3d.cs.colorado.edu/~l3d/systems/Ristretto/UserManual/fishtank/l

Figure 9: The Fish Tank—Created by a Community Using the Behavior Exchange

Our belief is that ideas of programming and simulation creation need to be reconceptualized from a social perspective. To support social perspectives such as communities of practice it is necessary to provide new forums for community members to share and exchange artifacts created. The Behavior Exchange (see Figure 9 in which a jointly constructed fish tank is displayed) is a forum that allows designers to exchange individual agents or entire simulations with other community members.

Figure 10 displays the essential processes underlying social creativity [Fischer & Scharff, 1998] supported by Visual AgenTalk and the Behavior Exchange in which two designers, Bob and Beth, share and further evolve their created artifacts. The processes are as follows:

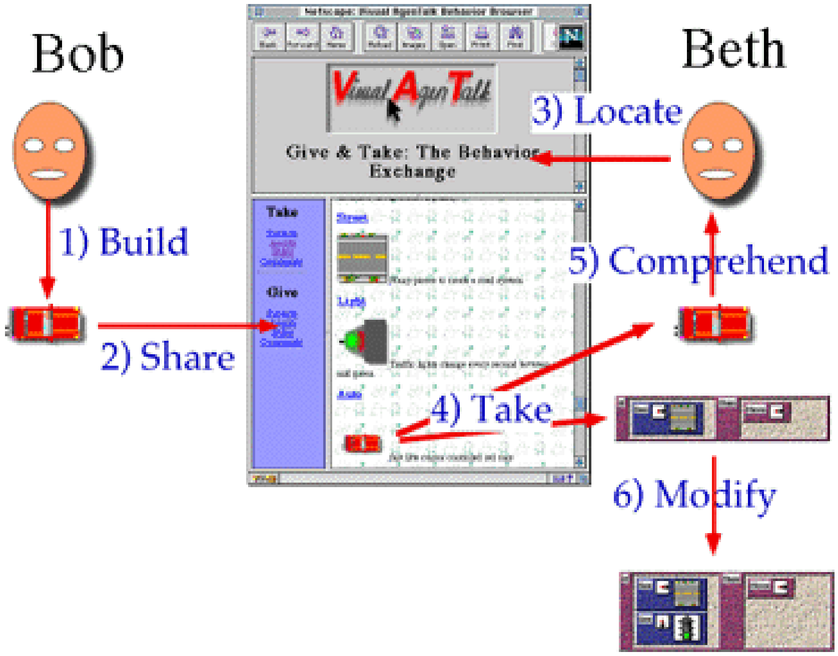


Figure 10: Processes Underlying the Behavior Exchange

1. ***Build***:—Bob builds a car agent for a traffic simulation
2. ***Share***:—Bob shares the car agent with the community
3. ***Locate***:—Some time later Beth looks for a suitable agent for her simulation and finds the car agent
4. ***Take:***—Beth takes the car agent simply by dragging it out of the Web page and directly dropping into her simulation
5. ***Comprehend:—***The car agent is not a closed black box. Beth can open the car up to understand how it works
6. ***Modify***:—Beth can modify the agent's look and behavior to make it fit her needs

Through the Behavior Exchange, the design of computational artifacts becomes a social activity. In previous attempts to increase the accessibility of Agentsheets, we supported the exchange of entire Agentsheets applications over the Web. The results indicated a need to exchange application components of a finer grain, including individual agents. The current Behavior Exchange features agents in categories such as transportation (roads, traffic lights, and cars), special effects (fire and flash), musical instruments (pianos and trumpets) and fish tanks (see Figure 9).

6. Assessment

Social creativity rests on the premise that one of the major roles for computational media is not to deliver existing and predigested information to individuals, but to provide the opportunity and resources for design activities embedded in social debates and discussions in which all people can act as designers rather than being confined to consumer roles [Fischer, 1998]. The SER model is an attempt to turn this vision more into a reality. It is motivated by how large software systems, such as Emacs, UNIX, Linux, and Microsoft Word, have evolved over time. In such systems, users develop new techniques and extend the functionality of the system to solve problems that were not anticipated by the system's authors (following the observation that any artifact should be useful in the expected way, but a truly great artifact lends itself to uses the original designers never expected; see Figure 1). New releases of the system often incorporate ideas and code produced by users.

DODEs pose a major additional challenge to make the SER model feasible and workable: whereas the people in the above-mentioned development environments are computationally sophisticated and experienced users, DODEs need to be extended by domain designers (end-users with respect to computational media) who are neither interested in nor trained in the (low-level) details of computational environments [Eisenberg & Fischer, 1994]. Domain designers are more interested in their design task at hand than in maintaining and evolving knowledge repositories per se. At the same time, important knowledge is produced during daily design activities that should be captured.

Computational support mechanisms are necessary prerequisites, but not sufficient conditions, to motivate people to become part of a “design culture.” People must be *motivated and rewarded* for investing time and effort to become knowledgeable enough to act as designers. These rewards may range from feeling in control (i.e., independent from “high-tech scribes”), being able to solve or contribute to the solution of a problem, a passion to master a tool in greater depth, a ego-satisfying contribution to a group, and/or good citizenship to a community.

Supporting social creativity along the dimensions discussed in this paper requires *computational media*. What are unique properties of computational media that are absent in principle from printed media? And why are these properties important social creativity? Printed media do not have interpretive power—they can convey information to us, but they cannot analyze the work products created by us. Critiquing for examples is a process that *analyzes our work products* and thereby increase the “back-talk” of an artifact by presenting a reasoned opinion about it. Computational media can make information relevant to the task at hand, thereby reducing the information overload problem or the need for decontextualized learning.

The approach presented in this paper is in line with characterizations of post-tayloristic work environments that are an attempt to increase the creativity of their workers by finding new ways to conceptualize working and learning as illustrated in Figure 11.

	traditional	**new**
Paradigm	Knowledge transmission	Knowledge construction
Learning	Classroom	On demand
Tasks	System driven (canonical)	User/task driven
Social structures	Individuals in hierarchical structures	Collaborative in flat structures (Communities of Practice)
Work style	Standardize	Improvise
Information spaces	Closed, Static	Open, Dynamic
Breakdowns	Error to be avoided	Opportunity for innovation and learning
Communication	Top-down	Peer-to-peer

Figure 11: Traditional versus New Models of Working and Learning in Organizations

7. Progress in the Last Decade

The 1998 Heron conference is the fourth in a series. The interest of researchers and practitioners in creativity has grown strongly during the last decade, as illustrated by the increasing number of books addressing this issue (for example, [Dartnall, 1994; Gardner, 1993; Gero & Maher, 1993], special issues of journals, and conferences). Computational environments to enhance human creativity continue to pose an interesting and exciting challenge for the scientific community. Of particular interest is the attempt to harvest community creativity rather than just individual creativity. Developments such as the Web provide the opportunity to employ the talent pool of a certain community of practice distributed all over the world. A misunderstanding often exists: namely, that electronic networks, being necessary to support social creativity, would also be sufficient.

From my personal point of view, progress has been specifically made in recognizing that the individual human mind is limited and that distributed cognition (distribution among humans and as well as humans and computational artifacts is required to cope with the challenging multi-disciplinary problems. By underdesigning systems (seeing them as living, growing entities rather than as finished products), the users are challenged to become designers and co-developers. Domain-oriented design environments have proven to be a valuable concept by providing a conceptual framework and innovative systems that allowed us to extend earlier approaches towards creativity by emphasizing the emerging possibilities to support social creativity in addition to individual creativity. The systems presented in this paper (e.g., the Envisionment and Discovery Collaboratory, DynaSites, Agentsheets, Visual AgenTalk, and the Behavior Exchange) provide support for distributed communities of practice to engage in collaborative design and social creativity. By extending the community of users, the amount of effort required by any individual is decreased, and at the same time the potential benefit to individuals is increased.

8. Conclusions

In this paper, I have articulated conceptual issues and described prototype systems that hopefully will contribute to understand the mutually constitutive relationship of the individual and the social [Brown & Duguid, 1992]. These frameworks and systems should contribute to resolve the contention that there is no individual creativity without social support and there is no social creativity without creative contributions of individuals. Increasing our understanding of distributed cognition, evolutionary and collaborative design, and the possibilities of new computational media will increase our possibilities to bring more social creativity into our lives.

Acknowledgments

The author would like to thank the members of the Center for LifeLong Learning & Design (L^3D) at the University of Colorado, who have made major contributions to the conceptual framework and systems described in this paper. I would like to acknowledge particularly: (1) Alexander Repenning and his collaborators, who developed Agentsheets, Visual AgenTalk, and the Behavior Exchange; (2) Ernie Arias, Hal Eden, Andy Gorman, and Eric Scharff, who developed the Envisionment and Discovery Collaboratory; (3) Jonathan Ostwald, who developed DynaSites; and (4) Kumiyo Nakakoji, who developed eMMa. All of them have helped me to gain a deeper understanding of creativity and its computational support.

The research was supported by (1) the National Science Foundation, Grants REC-9631396 and IRI-9711951; (2) Bell Atlantic/NYNEX Science and Technology Center, White Plains, NY; (3) Software Research Associates, Tokyo, Japan; (4) PFU, Tokyo, Japan; and (5) Daimler-Benz Research, Ulm, Germany.

References

Arias, E., Eden, H., Fischer, G., & Scharff, E. (1998) *Creating Shared Understanding through Collaborative Design with the Envisionment and Discovery Collaboratory*, University of Colorado (submitted to CHI ‘99).

Arias, E. G. (1995) “Designing in a Design Community: Insights and Challenges,” *Symposium on Designing Interactive Systems (DIS '95)*, pp. 259-263.

Boden, M. (1991) *The Creative Mind: Myths & Mechanisms*, Basic Books.

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.

Brown, J. S., Duguid, P. (1992) “Enacting Design for the Workplace,” In P. S. Adler & T. Winograd (eds.), *Usability: Turning Technologies into Tools,* Oxford University Press, Oxford, UK, pp. 164-197.

Bruner, J. (1996) *The Culture of Education*, Harvard University Press, Cambridge, MA.

Carroll, J. M., Rosson, M. B. (1987) “Paradox of the Active User,” In J. M. Carroll (eds.), *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, The MIT Press, Cambridge, MA, pp. 80-111.

Cross, N. (1984) *Developments in Design Methodology*, John Wiley & Sons, New York.

Csikszentmihalyi, M. (1990) *Flow: The Psychology of Optimal Experience*, HarperCollins Publishers, New York.

Dartnall, T. (Ed.) (1994) *Artificial Intelligence and Creativity*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Eisenberg, M., Fischer, G. (1994) “Programmable Design Environments: Integrating End-User Programming with Domain-Oriented Assistance,” *Human Factors in Computing Systems, CHI ‘94*, pp. 431-437.

Fischer, G. (1991) “Supporting Learning on Demand with Design Environments,” *International Conference on the Learning Sciences*, pp. 165-172.

Fischer, G. (1994a) “Domain-Oriented Design Environments,” *Automated Software Engineering*, 1(2), pp. 177-203.

Fischer, G. (1994b) “Turning Breakdowns into Opportunities for Creativity,” *Knowledge-Based Systems, Special Issue on Creativity and Cognition*, 7(4), pp. 221-232.

Fischer, G. (1998) “Beyond 'Couch Potatoes': From Consumers to Designers,” *3rd Asia Pacific Computer Human Interaction Conference*, pp. 2-9.

Fischer, G., Grudin, J., Lemke, A. C., McCall, R., Ostwald, J., Reeves, B. N., & Shipman, F. (1992) “Supporting Indirect, Collaborative Design with Integrated Knowledge-Based Design Environments,” *Human Computer Interaction, Special Issue on Computer Supported Cooperative Work,* 7(3), pp. 281-314.

Fischer, G., Lemke, A. C. (1988) “Construction Kits and Design Environments: Steps Toward Human Problem-Domain Communication,” *Human-Computer Interaction*, 3(3), pp. 179-222.

Fischer, G., Lemke, A. C., McCall, R., & Morch, A. (1996a) “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., Lindstaedt, S., Ostwald, J., Schneider, K., & Smith, J. (1996b) “Informing System Design Through Organizational Learning,” *International Conference on Learning Sciences (ICLS ‘96)*, pp. 52-59.

Fischer, G., McCall, R., Ostwald, J., Reeves, B., & Shipman, F. (1994) “Seeding, Evolutionary Growth and Reseeding: Supporting Incremental Development of Design Environments,” *Human Factors in Computing Systems (CHI ‘94)*, pp. 292-298.

Fischer, G., Nakakoji, K. (1992) “Beyond the Macho Approach of Artificial Intelligence: Empower Human Designers—Do Not Replace Them,” *Knowledge-Based Systems Journal*, 5(1), pp. 15-30.

Fischer, G., Nakakoji, K. (1994) “Amplifying Designers' Creativity with Domain-Oriented Design Environments,” In T. Dartnall (eds.), *Artificial Intelligence and Creativity*, Kluwer Academic Publishers, The Netherlands, pp. 343-364.

Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., & Sumner, T. (1993) “Embedding Critics in Design Environments,” *The Knowledge Engineering Review Journal*, 8(4), pp. 285-307.

Fischer, G., Scharff, E. (1998) “Learning Technologies in Support of Self-Directed Learning,” *Journal of Interactive Media in Education*, 1998(4), www-jime.open.ac.uk/98/4 (electronic journal).

Galegher, P., Kraut, R., & Egido, C. (Ed.) (1990) *Intellectual Teamwork*, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ.

Gardner, H. (1993) *Creating Minds*, Basic Books, Inc, New York.

Gero, J. S., Maher, M. L. (1993) *Modeling Creativity and Knowledge-Based Creative Design*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Greenbaum, J., Kyng, M. (1991) *Design at Work: Cooperative Design of Computer Systems*, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ.

Grudin, J. (1991) “Interactive Systems: Bridging the Gaps Between Developers and Users,” *Computer*, 24(4), pp. 59-69.

Grudin, J. (1994) “Computer-Supported Cooperative Work: History and focus,” *IEEE Computer*, 27(5), pp. 19-26.

Henderson, A., Kyng, M. (1991) “There's No Place Like Home: Continuing Design in Use,” In J. Greenbaum & M. Kyng (eds.), *Design at Work: Cooperative Design of Computer Systems*, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, pp. 219-240.

Illich, I. (1973) *Tools for Conviviality*, Harper and Row, New York.

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. (1991) “Situated Learning in Communities of Practice,” In L. Resnick, J. Levine, & S. Teasley (eds.), *Perspectives on Socially Shared Cognition*, American Psychological Association, Washington, DC, pp. 63-82.

Lindstaedt, S. (1998) *Group Memories: A Knowledge Medium for Communities of Interest*, Ph.D. Dissertation, University of Colorado at Boulder.

Nakakoji, K. (1993) *Increasing Shared Understanding of a Design Task Between Designers and Design Environments: The Role of a Specification Component*, Ph.D., University of Colorado at Boulder.

Nakakoji, K. (1998) *Workshop on "Collective Creativity in Design through Concept Externalization" (CCDCE),* At http://hawk.aist-nara.ac.jp/CCC/sakigake/.

Nakakoji, K., Aoki, A., & Reeves, B. N. (1996) "Knowledge-Based Cognitive Support for Multimedia Information Design," *Information and Software Technology*, 38(3), pp. 191-200.

Nardi, B. A. (1993) *A Small Matter of Programming*, The MIT Press, Cambridge, MA.

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

Ostwald, J. (1997) *DynaSites,* At http://www.cs.colorado.edu-/~ostwald/home.html.

Pea, R. (1996) "Seeing What We Build Together: Distributed Multimedia Learning Environments for Transformative Communications," In T. Koschmann (eds.), *CSCL: Theory and Practice of an Emerging Paradigm*, Lawrence Erlbaum Associates, Inc., Mahwah, NJ, pp. 171-186.

Petroski, H. (1985) *To Engineer Is Human: The Role of Failure in Successful Design*, St. Martin's Press, New York.

Polanyi, M. (1966) *The Tacit Dimension*, Doubleday, Garden City, NY.

Popper, K. R. (1965) *Conjectures and Refutations*, Harper & Row, New York, Hagerstown, San Francisco, London.

Raymond, E. S. (1998) *The Cathedral and the Bazaar*, At http://earthspace.net /~esr/writings/cathedral-bazaar/cathedral-bazaar.html.

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.

Repenning, A. (1997) *Agentsheets,* At http://www.cs.colorado.edu/~l3d/systems /agentsheets/.

Repenning, A., Ambach, J. (1997) "The Agentsheets Behavior Exchange: Supporting Social Behavior Processing," *Computer-Human Interaction (CHI '97)*, pp. 26-27 (Extended Abstracts).

Resnick, L. B., Levine, J. M., & Teasley, S. D. (Ed.) (1991) *Perspectives on Socially Shared Cognition*, American Psychological Association, Washington, D.C.

Resnick, M. (1994) *Turtles, Termites, and Traffic Jams*, The MIT Press, Cambridge, MA.

Rittel, H. (1984) "Second-Generation Design Methods," In N. Cross (eds.), *Developments in Design Methodology*, John Wiley & Sons, New York, pp. 317-327.

Rogoff, B., Lave, J. (1984) *Everyday Cognition*, Harvard University Press, Cambridge, MA.

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

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

Shipman, F. (1993) *Supporting Knowledge-Base Evolution with Incremental Formalization*, Ph.D., Department of Computer Science, University of Colorado at Boulder.

Simon, H. A. (1996) *The Sciences of the Artificial, 3rd edition*, The MIT Press, Cambridge, MA.

Terveen, L. G., Selfridge, P. G., & Long, D. M. (1995) "Living Design Memory: Framework, Implementation, Lessons Learned," *Human-Computer Interaction*, 10, pp. 1-37.