

# Knowledge Management: Problems, Promises, Realities, and Challenges

Gerhard Fischer and Jonathan Ostwald, *University of Colorado*

Those who cannot remember the past are condemned to repeat it. —George Santayana  
 Innovation is everywhere; the difficulty is learning from it. —John Seeley Brown

*The authors' knowledge management approach assumes that knowledge is not a commodity but that it is collaboratively designed and constructed.*

**T**he first quote reflects the motivation underlying traditional knowledge management (KM), in which the goal is to store information from the past so that lessons will not be forgotten. This perspective implies that future information needs will be the same as past needs. Consequently, this perspective treats knowledge

workers as passive recipients of information.

The second quote more closely reflects a *design perspective*<sup>1</sup> of knowledge management. In this perspective, knowledge workers constantly create new knowledge as they work. KM's goal is to enable innovative practice at an organizational (community) level by supporting collaboration and communication among knowledge workers in the same domain and across domains.

This article explores the design perspective's implications for KM. We examine the major problems our approach must address, the promises it offers, the realities we have explored in our work, and the continuing challenges. Table 1 summarizes the article's key ideas.

## A basic framework

KM is a cyclic process involving three related activities: creation, integration, and dissemination (see Figure 1).

In this model, computation supports human knowledge activities by manipulating information. An information repository stores information that was created in the past and is disseminated throughout an organization or group. We can classify KM

approaches according to how they perform these basic activities. For example, different approaches might store different kinds of information, support different people to create information, or employ different mechanisms and strategies to disseminate information.

In traditional KM approaches, management collects and structures an organizational memory's contents as a finished product at design time (before the organizational memory is deployed) and then disseminates the product. Such approaches are top-down in that they assume that management creates the knowledge and that workers receive it.

Our design perspective is an alternative that relates working, learning, and knowledge creation. In this framework, workers are reflective practitioners,<sup>2</sup> who struggle to understand and solve ill-defined problems. Learning is intrinsic to problem solving, because problems are not given but must be framed and solved as a unique instance. This perspective has two essential aspects. First, workers, not managers, create knowledge at use time. Second, knowledge is a side effect of work. Table 2 compares the traditional KM perspective with our perspective.

Table 1. Article overview.

	Knowledge creation	Knowledge integration	Knowledge dissemination
Key idea	Knowledge is a work product, not an existing commodity.	Workers integrate new knowledge into repositories at use time.	Workers get information in the context of work, not in the classroom.
Problems	<i>Creating shared understanding</i> Externalizations create shared understanding.	<i>Putting communities in charge</i> Users must be empowered to manage their own information (and environments).	<i>Alleviating information overload</i> The limiting resource for knowledge work is not information but attention.
Promises	<i>Social creativity</i> Workers are informed participants in the creation of knowledge, not consumers of prepackaged information.	<i>Living organizational memory</i> Information repositories are evolved by unself-conscious cultures of design.	<i>Attention economy</i> Information is delivered to workers when it is relevant to their specific needs.
Realities	<i>Envisionment and Discovery Collaboratory</i> Boundary objects support communities of interest to build shared understanding.	<i>DynaSites</i> Open information spaces are evolved by users with system support for integration.	<i>Domain-Oriented Design Environments</i> Design tools and information repositories are integrated to enable knowledge delivery.

**Creation**

KM approaches exist because work is increasingly information intensive. Traditional KM approaches assume that the critical issue for workers is to find the “answers” in organizational memory that apply to the current problem. A design-based approach assumes that the organizational memory will not contain all the knowledge required to understand and solve such problems. So, workers must create new knowledge.

**Integration**

In the design perspective, an organizational memory plays two roles. First, it is a source of information to help workers understand the problems they face. Second, it is a receptacle for new information and products created during work. In traditional KM approaches, knowledge engineers carefully craft a knowledge base that will periodically be updated. In a design-based approach, organizational memory is a continuously evolving information space that is fed directly by the knowledge created during work. So, information repositories and organizational memories are

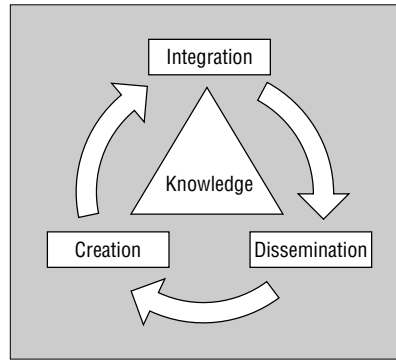


Figure 1. Knowledge management as a cyclic process.

not huge, impenetrable “write-only” stores. They are actively integrated into the work process and social practices of the community that constructs them.

Although the problems workers solve are unique in some aspects, they are also similar to those previously solved. The challenge for knowledge integration is to make the connections between old and new knowledge so

that the organizational memory improves its ability to inform work. In the traditional KM approaches, this was the knowledge engineer’s job. In a design-based approach, users do it at use time.

Knowledge integration comprises two tasks:

- *Conceptual generalization*—relating information from one context to information from another.
- *Representational formalization*—putting information in a form such that computational mechanisms can access and interpret it.

Both tasks require effort beyond what most workers consider their core responsibility. Conceptual generalization requires an understanding of the domain, while formalization requires the ability to map from domain concepts into the formalizations the system requires. A major concern for our design-based approach is to capture information from the work process without extra effort by the users and then to help them formalize

Table 2. Two perspectives on knowledge management.

	Traditional perspective	Our perspective
Creation	Specialists (for example, knowledge engineers)	Everyone (for example, people doing the work), collaborative activity
Integration	At design time (before system deployment)	At use time (an ongoing process)
Dissemination	Lecture, broadcasting, classroom, decontextualized	On demand, integration of learning and working, relevant to tasks, personalized
Learning paradigm	Knowledge transfer	Knowledge construction
Tasks	System driven (canonical)	User or task driven
Social structures	Individuals in hierarchical structures, communication primarily top-down	Communities of practice, communication primarily peer-to-peer
Work style	Standardize	Improvise
Information spaces	Closed, static	Open, dynamic
Breakdowns	Errors to be avoided	Opportunities for innovation and learning

the captured information incrementally, rather than at the time of capture.<sup>3</sup>

## Dissemination

This activity makes information in the organizational memory available to workers to help their problem solving. Traditional KM approaches assume that workers perform repetitive and predictable tasks, so they disseminate knowledge through classroom training or printed reference documents. These approaches separate learning and working. They typically use information technology to broadcast information (for example, email) or to provide searchable databases. As we mentioned earlier, the information that workers receive or access comes from management (or the creator of the training materials) rather than from coworkers.

In the design perspective, the specific information needs of workers are unpredictable. The need for information results from particular situations that arise from a worker's struggle to understand a problem. The context of problem solving dictates the information demand and provides the context for learning. On-demand information integrates working and learning, because the need for learning comes from work, and the learning takes place within the context of the work situation.

## Problems

Now we look at some of the major barriers to implementing a design-oriented approach.

## Creating shared understanding

KM aims to increase the ability of workers to perform knowledge-intensive tasks. From the perspective of work as creative design, we can restate this purpose simply as understanding the problem at hand.

*External representation.* An important aspect of design is the creation of artifacts that externalize knowledge. This is important for three reasons:

- In so doing, we begin to move from vague, tacit conceptualizations of an idea to a more explicit representation.
- The artifact provides a means for others to interact with, react to, negotiate around, and build on the externalized idea.
- The artifact provides an opportunity to create a common language of understanding.

We have found that using external representations exposes, and focuses discussion on, relevant aspects of the framing and understanding of the problem being studied, such as tacit attitudes, values, and perspectives.<sup>4</sup> This is because designers engage in a “conversation with the materials of a situation.”<sup>2</sup> In this conversation, designers interact with an external representation of the problem, and the situation talks back to them, causing breakdowns in their prior understandings. To designers, breakdowns are not mistakes but opportunities to create new understandings. When a breakdown occurs, designers reflect on the breakdown,<sup>2,5</sup> learning more about the problem, its framing, and possible solutions.

Closed systems do not give communities control over their own knowledge but put a gulf between creation and integration. So, innovations happen outside systems.

*Collaborative design.* Increasingly, groups or communities working together—not individuals—perform design tasks. Complexity in collaborative design arises from the need to synthesize different perspectives of a problem, to manage large amounts of information relevant to a design task, and to understand the design decisions that have determined a designed artifact's long-term evolution.

Our work focuses on two types of groups, *communities of practice*<sup>6</sup> and *communities of interest*.<sup>4</sup>

Communities of practice consist of people sharing a common practice or domain of interest. CoPs are sustained over time. They provide a means for newcomers to learn about the practice and for established members to share knowledge about their work and to collaborate on projects. They need support for understanding long-term evolution of artifacts and for understanding problems caused by rapid change in their domain.

CoIs consist of people from different fields who come together to work on a particular

project or problem. They typically exist for a project's duration. They need support for creating shared understanding among stakeholders from different backgrounds, who bring different perspectives and languages to the problem.

CoPs are typically associated with KM. CoIs are becoming increasingly involved in KM as projects become more interdisciplinary and as collaborative design brings together specialists from many domains. The challenge is to meaningfully bridge and integrate these various perspectives. Such integration requires support for *reflection in action*. For collaborative design, where many people work together to understand a problem, design becomes a conversation in a more literal sense. That is, external representations facilitate a conversation not only with the design situation but also with other designers. In this way, externalizations expose breakdowns due to a lack of understanding of the problem, conflicts among perspectives, or the absence of shared understanding. As we mentioned earlier, such breakdowns are opportunities to build new, shared understandings.

Collaborative designs result in work products<sup>7</sup> that are enriched by the multiple perspectives of the participants and the discourses that result from the process. This integrates the individual and the group knowledge in ways impossible in settings that rely solely on “divide and conquer” team organization.

## Putting communities in charge

The view of workers as reflective practitioners within CoPs does not correspond to what is taught in training or what is contained in information systems supporting the traditional KM view. Traditional information systems are closed systems that store answers to questions that might arise during work, under the assumption that workers are performing tasks that have been anticipated and described. This assumption is a barrier to innovation, because it does not let workers share their new ideas for their peers to discuss, debate, or build on. Closed systems do not give communities control over their own knowledge but put a gulf between creation and integration. So, innovations happen outside systems, and systems contain information that is chronically out of date and that reflects an outsider's view of work.<sup>1</sup>

*Capturing information at use time.* Design communities have learned that anticipating all possible uses at design time (that is, when

the system is created) is impossible.<sup>8</sup> Skilled domain professionals will change their work practices over time, and new information will become available. If users cannot modify a system at use time to support new practices and new emerging information, they will be locked into old patterns of use, or they will abandon the system for one that better supports how they want to work.

An example of a successful open system that allows user modifications is Xerox's Eureka. It is an information repository for copier repair representatives that the company believes has saved up to \$100 million a year.<sup>1</sup> The users create and evolve the repository's information, subject to peer review. Eureka represents an early approach at taking bottom-up knowledge creation seriously, in which users gain peer recognition through their contributions to the system.

The Eureka system takes an explicit approach to knowledge capture; it demands a fair amount of work by users to input their problem-solving knowledge. This is because much of the Eureka user's work takes place outside the KM system and must be input later.

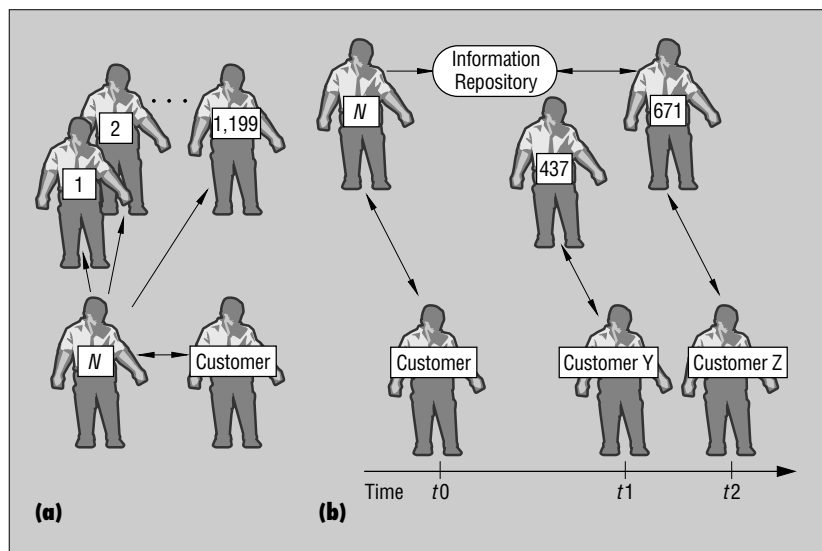
*Integrating tools and repositories.* Systems that integrate work tools with information repositories can support more subtle information capture. For example, *social navigation*<sup>9</sup> and *recommender systems* collect information in the background as users do their work and then provide this information to a wider community to inform their decision making.

These approaches advance Vannevar Bush's "trailblazer" concept and Will Hill and James Hollan's "read ware and edit ware" concept<sup>10</sup> to make these unique contributions:

- Traces are not preplanned aspects of a space, but rather are "grown" (or created dynamically) in a more organic, or bottom-up, fashion.
- They provide information that reflects what people actually do rather than what system designers think people should be doing.
- They rely on how people occupy spaces and transform them by leaving their marks on them.
- They often rely on spatial metaphors (drawing on work in architecture and urban design).

### Alleviating information overload

The KM community has focused consid-



**Figure 2. Two help desk scenarios: (a) broadcasting decontextualized information can cause information overload; (b) access to contextualized information supports learning on demand.**

erable effort toward codifying and storing knowledge. Knowledge workers are now routinely equipped with documents such as user manuals and online help systems that contain thousands of pages of information. Reading these documents from beginning to end is a waste of time. Much information will not make sense in the abstract, and workers will have forgotten the information by the time it becomes necessary.

*Attention, please.* The scarcest resource for most of us as we try to understand and solve problems is not information; it is *attention*.<sup>11</sup> Herbert Simon said,

If computers are to be helpful to us at all, it must not be in producing more information—we already have enough to occupy us from dawn to dusk—but to help us to attend to the information that is the most useful or interesting or, by whatever criteria you use, the most valuable information.

As this quote implies, we have more information available than we have attention to understand and apply it. At the same time, finding information relevant to the task at hand is becoming increasingly critical.

To address information overload, KM approaches must provide the information workers need, when they need it. The following example illustrates the limitations of traditional approaches to knowledge dissemination.

*More than just access.* One of our collaborating companies employs 1,200 help desk people, who help customers solve problems

over the phone. Suppose desk person *N* expends considerable effort to solve a customer's difficult problem, generating new knowledge in the process. How should this knowledge be documented and shared with the other help desk people? Should *N* broadcast (for example, by emailing to a companywide list) this problem and its solution to the 1,199 other help desk people, as Figure 2a illustrates?

We believe the answer is no. In general, this information will not be relevant to the other help desk people at the same point in time. All these people (like most knowledge workers) suffer not from a scarcity of information but from information overload. The problem will worsen if a help desk person receives more decontextualized information that appears irrelevant.

Figure 2b illustrates a more promising strategy. The problem-solving knowledge that *N* created and documented is captured in an organizational memory. In the future, when a help desk person encounters a problem in which *N*'s solution is relevant, the information is available—if the desk person can find it.

The standard KM approaches for knowledge dissemination are access approaches that let users search for stored information using database queries. Although such approaches are necessary for locating information, they are not always sufficient. For example, users might not be able to articulate their information needs in a way that the access mechanisms require. Also, users might not be motivated to search for information if they don't know that relevant infor-

mation exists. Or, users might not be aware of the need for information in the first place.

Design-oriented KM approaches must go beyond the traditional forms of knowledge dissemination if they are to address information overload. The help desk example represents these core technical issues for KM environments:

- devising computationally tractable representations of experiences,<sup>3</sup>
- developing retrieval technologies that recognize complex as well as surface similarities,<sup>12</sup>
- capturing significant portions of knowledge that practitioners generate in their work,<sup>13</sup> and
- nurturing a culture that motivates individuals to work for the good of the group or organization.<sup>14</sup>

## Promises

We differentiate between two types of promises. The first constitutes myths, for which little evidence exists, and which might lead us to work toward questionable goals. Although KM's promise is exciting and real, misconceptions exist that we must expose and examine. KM shares the hype and unrealistic expectations that have surrounded other disciplines such as expert systems and object-oriented design.<sup>15</sup> For example, people assumed that these technologies by themselves would do the job.

The second type of promises offers alternatives to these myths. These promises focus on the three basic KM activities: knowledge creation in the context of *social creativity*,<sup>16</sup> knowledge integration in the context of *living information repositories*,<sup>17</sup> and knowledge dissemination in the context of an *attention economy*.<sup>1</sup>

## Social creativity

The first myth is that knowledge is a commodity. This myth has two parts:

First, we can simply and explicitly “capture” the knowledge of a 30-year expert. So, we can fire the expert and hire someone with no relevant skills off the street who can now use the “knowledge base” to perform like an expert.

Second, in the ideal company, information technology will capture all knowledge worldwide and instantly feed it through high bandwidth lines to a central location. At this location, experts will make globally optimal decisions for the entire company and feed them back to the periphery for implementation.<sup>18</sup>

*Knowledge > information.* John Brown and Paul Duguid<sup>1</sup> argue convincingly that knowledge is more than just information because it

- usually entails a knower,
- appears harder to detach than information, and
- is something what we digest rather than merely hold.

A consequence of these observations is that attention to knowledge (rather than just to information) requires attention to people, including their tasks, motivation, and interests in collaboration. Knowledge is information that is attached to a particular context (for example, a task, problem, or question).

The heart of intelligent human performance is not the individual human mind but groups of minds interacting with each other and with tools and artifacts.

Although information can be easily transmitted from place to place and person to person, the underlying context cannot. Information technology is necessary to realize the KM cycle of creation, integration, and dissemination, but technology alone is insufficient. KM requires changing work practices and attitudes to acknowledge the importance of the knowledge worker and the contexts of work in transforming information into capability for effective action.<sup>15</sup>

## *Social creativity and informed participation.*

The heart of intelligent human performance is not the individual human mind but groups of minds interacting with each other and with tools and artifacts. Social creativity grows out of the relationship between an individual and the world of his or her work, and out of the ties between an individual and other human beings. The knowledge to understand, frame, and solve most design problems does not preexist but is constructed and evolves during problem solving. So, instead of sim-

ply providing better access, KM should support *informed participation*<sup>1</sup> in collaborative problem-solving processes and communities.

However, informed participation is impossible in communities that strictly separate users from designers and developers. This separation is undesirable and unproductive.

Users must acquire a new mind-set—they are no longer passive receivers of knowledge but are active researchers, constructors, and communicators of knowledge. Knowledge is no longer handed down from above; it is constructed collaboratively in the contexts of work.

## Living organizational memories

The second myth is that the evolution of complex artifacts and information spaces can be purely self-organized (decentralized). KM can learn some lessons from open-source development projects,<sup>19</sup> which always have a core set of project leaders who have the final say on what course a project's evolution takes. These people centrally integrate information that others have contributed in a decentralized manner. Contributors are explicitly acknowledged and often assume responsibility for their subsystem's evolution. Open-source projects have many varieties of control structures, but each project will have some centralized responsibility.<sup>20</sup> No project practices purely decentralized evolution.

The evolution of open KM systems must also have elements of decentralized evolution and centralized integration. The mix of these modes and the means of selecting individuals to assume responsibility will take many forms. Later in this article, we present a general framework identifying essential activities and roles for sustained evolution of open systems. A major difference between open-source projects and open KM systems is that the latter's users are end users.

The goal of making user-modifiable systems does not imply transferring the responsibility of good system design to the user. Normal users generally will make poorer modifications than a system specialist would. Users are not concerned with the system per se but with doing their work. On the other hand, users are concerned with the system's adequacy as a tool for their work. So, they experience how the tool's capabilities fit, or do not fit, their needs. This is knowledge the specialist cannot have, because the specialist does not use the tool to do work. User-modifiable systems let the user adapt a system directly, without requiring a specialist and

**Table 3. Information repositories evolved by specialists versus those evolved by knowledge workers.**

	<b>Evolved by specialists</b>	<b>Evolved by knowledge workers</b>
Examples	ACM digital library	Web sites of communities of practice; Eureka
Nature of individual entries	Database-like entries	Narratives and stories
Economics	Requires substantial extra resources	An additional burden on the knowledge workers
Delegation	Possible in domains in which entries or objects are well defined	Performed by problem owners, because the entries or objects are emerging products of work
Design culture	Self-conscious	Unself-conscious
Motivation	Work assignment	Social capital

without requiring deep knowledge of the system’s inner workings.<sup>21</sup> Designing such systems does not decrease the system specialist’s responsibility or importance. It shifts the responsibility from designing a finished system at design time to designing a system that the user can adapt and modify at use time.

*A proper evolution.* Living organizational memories offer these promises and opportunities:

- They are information spaces owned by the people and communities who use them to do work, not by management or the IT department.
- They support the collaborative and evolutionary design of complex systems by providing a means to integrate the many contributions of many people.
- They are open and evolvable systems, serving not only as information repositories but also as mediums of communication and innovation.
- They can evolve through many small contributions by many people rather than through large contributions by a few people (as has been the case for previous knowledge-based systems).

*Unself-conscious cultures of design.* When an artifact’s users can recognize and repair breakdowns as they use it, they are empowered to maintain the artifact’s fit to its changing environment. The architect and design methodologist Christopher Alexander wanted his buildings to be continually maintained and enhanced in this manner by the people who inhabited them. He coined the phrase *unself-conscious culture of design*<sup>22</sup> to describe this form of design-in-use. In unself-conscious design, breakdown and correction occur side by side; no formal set of rules describes how to repair breakdowns, because the breakdowns

are unanticipated. Instead, the knowledge to repair breakdowns comes from the user, who can best recognize the lack of fit and how to change the artifact to improve its fit.

In an unself-conscious culture of design, an artifact’s failure or inadequacy leads directly to an action to change or improve it. For example, when a house’s owner is also its builder, constant rearrangement of unsatisfactory details are possible. In KM, open systems put the owner of problems in charge. Because the owners are in charge, the positive elements of the unself-conscious design culture can be exploited in the evolution of organizational memories. In such environments, the end users, not the system builders, experience breakdowns. These breakdowns lead the users to continually and directly evolve and refine their information space, without relying on professionals.

Sustaining the usefulness and usability of living information repositories over time involves important challenges and trade-offs (summarized in Table 3). These trade-offs depend on whether these information repositories are evolved by specialists or by knowledge workers.

**Attention economy**

The third myth is that “anytime and anywhere” information access will solve KM problems. Because we believe that the scarcest resource for most people is attention, we claim that the real challenge is to “say the right thing at the right time in the right way.” This is possible only with computational environments that take the user’s context into account (for example, what the users are doing, what they know, where they are, and what have they done). KM needs to exploit computational media’s capabilities for interpreting information to support attention economies, in which attention is the most valued resource.

*Unique capabilities of computational media.* Printed media do not have interpretive power—they can convey information, but they cannot analyze the work products we create. Computational media can provide

- information relevant to the task at hand,<sup>23</sup> thereby reducing the information overload or the need for decontextualized learning, and
- the foundation for on-demand information, detail, and learning.

*Beyond access approaches.* Information delivery complements information access approaches for disseminating information. While information access is a user-initiated search, information delivery is a system-initiated presentation of information intended to be relevant to the user’s task. Table 4 compares information access and delivery technologies.

Support for information access is indispensable because designers must be able to search for needed information. The ability of information access technologies to retrieve information related at levels beyond surface similarities has improved. However, they remain limited in principle because their users must articulate information needs.

Information delivery technologies exploit the scarce resource of attention better, because they infer a user’s information needs rather than requiring the user to explicitly formulate a query. Information delivery is particularly important when designers are not motivated to look for information or when they are not aware of the need for information in the first place.

To deliver information relevant to the user’s task, delivery mechanisms face two major challenges:

- *Determining the user’s information needs.*

Table 4. Comparison of information access and delivery approaches.

	Access	Delivery
Examples	Passive help systems, browsing, Web search engines, bookmarks	Microsoft's "Tip of the Day," broadcast systems, critiquing, active help systems, agent-based systems
Strengths	Nonintrusive, user controlled	Serendipity, creating awareness for relevant information, rule enforcement
Weaknesses	Task-relevant knowledge might remain hidden because the user couldn't specify it in a query	Intrusiveness, possibility of decontextualized information
Major system design challenges	Supporting users in expressing queries, better indexing and search algorithms	Context awareness (intent recognition, task models, user models, relevance to a task)

Information needs can be inferred from the task at hand (what the user is doing and the actions he or she has performed) and from the user's intentions. Determining the task at hand is challenging, but inferring intentions is even more difficult. Delivery mechanisms must operate with incomplete information about the designers' intentions because they are not completely known, even by the designer.

- *Intervention strategies.* Although delivery mechanisms can be designed and tailored for minimum disruption, a conflict will always arise between the need to inform users and the desire not to inundate them with irrelevant messages.

These are formidable challenges. However, we believe that information delivery will fulfill the promise of information on demand, thereby realizing the vision of an attention economy.

**Realities**

Over the last decade of research on our integrated KM approach, we have developed

- *conceptual frameworks*, such as the *seedling*, *evolutionary growth*, *reseeding* (SER) process model; the integration of information access and information delivery; *boundary objects*; and *courses-as-seeds*; and
- *prototype systems* such as Domain-Oriented Design Environments, the Envisionment and Discovery Collaboratory, and DynaSites to validate and extend the frameworks.

Table 5 summarizes these efforts. Now we look at how this work addresses the problems and promises we've discussed.

**Domain-Oriented Design Environments**

DODEs are a class of integrated systems

that uniquely support the KM cycle. We have built DODEs in many domains. During this process, we have developed a domain-independent software architecture describing the tools and knowledge-based mechanisms that support the KM processes.<sup>5</sup> We now examine NetDE, a DODE that supports the creation and management of knowledge in the domain of local area network design and administration (see Figure 3).

To create LAN layouts, users employ a *construction worksheet* (see Figure 3a and 3b), in which they locate network devices and connect them using different cables and network protocols. They can use a *simulation component* to visualize dynamic behaviors as they make changes and try new ideas. A *specification component* (see Figure 3c) lets users articulate high-level intentions for their project that are not explicit in the worksheet, such as a ranking of priorities.

The NetDE information repository consists of a *group memory* and a *catalog*. The group memory (see Figure 3d) holds information from previous projects, email communication archives, and other textual information. The catalog (see Figure 3e) contains example networks. Knowledge workers can use them to see how a similar problem was solved, to understand the evolution of a net-

work being designed, or as a starting point for a new design.

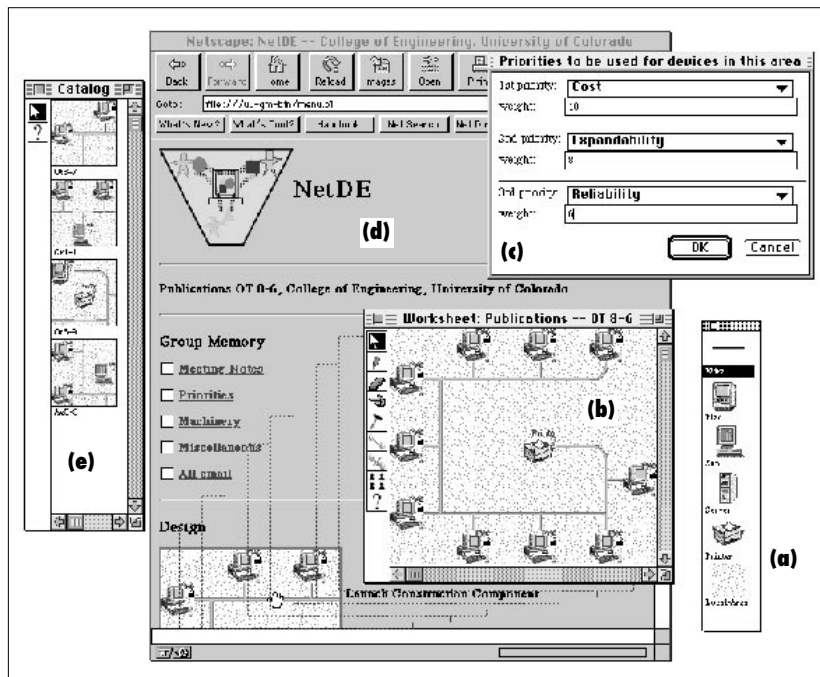
Like most KM environments, NetDE supports information access through searching and browsing. Unlike most KM environments, NetDE can play an active role in knowledge dissemination. Critiquing mechanisms (*critics*) monitor the actions of users as they work and inform them about potential problems. Users can elect to see information relevant to a problem. If they do, critics place the user in the repository where relevant information is located. The user can then browse the proximity to learn more about the problem. In this way, NetDE integrates information access and delivery approaches.

Critics exploit the context defined by the state of the construction worksheet and the simulation and specification components to identify potential problems and to determine what information to deliver. This context enables precise intervention by critics, reduces annoying interruptions, and increases the relevance of information delivered to designers.

Critics embedded in design environments increase the user's understanding of problems to be solved, point out information needs that might have been overlooked, and locate relevant information in large information spaces. Embedded critics save users the trouble of explicitly querying the system for

Table 5. Our conceptual contributions and prototype systems.

Area	Contribution	Example
Creation	Boundary objects (supporting informed participation); seeding, evolutionary growth, reseeding process model	Envisionment and Discovery Collaboratory
Integration and evolution	Collaborative, decentralized, evolvable information spaces	DynaSites
Dissemination and learning	Information delivery (learning on demand, specification components, using an artifact as a query)	Domain-Oriented Design Environments



**Figure 3. A Domain-Oriented Design Environment for designing local area networks: (a–b) the construction worksheet; (c) the specification component; (d) the group memory; (e) the catalog.**

information. Instead, the design context serves as an implicit query. Rather than specifying information needs, the user only has to click on a critiquing message to obtain relevant information.

### The Envisionment and Discovery Collaboratory

For our first-generation DODEs, we simplified the process of “context awareness,” because all activities happened inside the computational environment rather than in the external world. The *Envisionment and Discovery Collaboratory*<sup>4</sup> represents second-generation DODEs that support social interaction by creating shared understanding among various stakeholders, contextualizing information to specific tasks, and creating boundary objects as externalizations in collaborative design activities. The EDC extends the original DODE approach by integrating computational environments and (computationally enriched) external physical worlds with mechanisms capturing the larger (often unarticulated) context of what users are doing. DODEs primarily support CoPs; the EDC also supports CoIs.

*Supporting CoIs.* The EDC provides objects that all stakeholders can understand and manipulate. It also provides underlying computational support for trying out alternative solutions, accessing relevant information,

and capturing information and design rationale from the design process.

Stakeholders using the EDC convene around a computationally enhanced table that serves as the *action space*. Currently realized as a touch-sensitive surface, the action space

lets users manipulate a computational simulation projected on the surface by interacting with physical objects placed on the table. The simulation is an interactive model of the design problem that reacts to the user’s input. It lets users explore alternative solutions in a potentially complex design space. Flanking the table is another touch-sensitive (vertical) surface that serves as the *reflection space*. The reflection space displays information that is relevant to the context as defined by the simulation.

The EDC framework is applicable to different domains, but our initial effort has focused on urban planning and decision making, specifically in transportation planning and community development. In Figure 4, neighbors are filling out a Web-based transportation survey associated with the simulation being constructed.

*Boundary objects.* Action space objects are domain oriented—they look and behave like objects in the problem domain. These objects and behaviors are meaningful to all stakeholders who are familiar with the domain. However, the stakeholders might not share the precise meanings of the objects and the implications of the meanings for design decisions. The objects serve as boundary objects by providing a common starting ground for stake-



**Figure 4. The Envisionment and Discovery Collaboratory. In the *action space* (foreground), stakeholders use physical objects to interact with an underlying computational simulation environment. In the *reflection space* (background), stakeholders access information, fill out surveys, and add new information.**



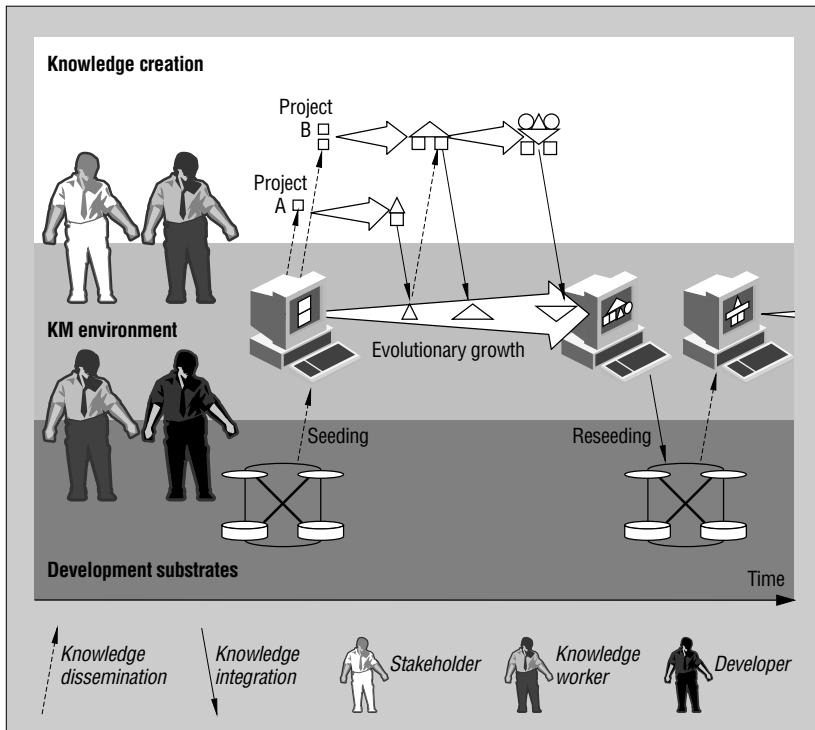


Figure 5. Knowledge management processes in the context of the *seeding, evolutionary growth, reseeding* model. A KM environment (including its content) starts in the seeding phase. The evolutionary growth phase occurs as people use the environment on projects. Occasionally, the information integrated during this growth requires a reseeding phase. Evolutionary growth then continues until reseeding is again necessary.

holders to identify and explore the differences in their understandings and to build new understandings that bridge the boundaries.

For example, in the transportation-planning domain, stakeholders include transportation engineers and neighborhood residents who will work together to improve the design of bus routes in their neighborhood. In the action space, they use domain objects such as buses, bus stops, neighborhoods, and streets to explore the problem's different facets. An engineer might think of a bus stop in terms of its capacity to serve a certain-size neighborhood, while a resident might think of a bus stop in terms of its convenience to his or her house or in terms of its safety at night. The bus stop object in the EDC is a boundary object for engineers and residents to build a shared understanding of the "bus stop" concept in terms of the importance and implications for the particular design. The action space simulation, which helps stakeholders explore alternatives, and the reflection space, which provides background information about each perspective, enhance this process.

### The seeding, evolutionary growth, reseeding process model

We developed the SER process model to understand the balance between centralized and decentralized evolution in sustained development of large systems. Our goal is to apply lessons learned from successes such as open-source software to domains and communities, such as KM, that have not traditionally been viewed from this perspective.

The SER model situates the KM cycle in a larger context by addressing how to initiate and sustain it (see Figure 5). The model describes three phases of evolution in terms of the stakeholders involved and their activities. The *seeding* phase creates the initial conditions for the KM cycle. The cycle's activities are the driving force of the *evolutionary-growth* phase. Finally, *reseeding* is a periodic effort to organize and tune the KM environment.

*Seeding.* In this phase, system developers and users work together to develop an initial KM environment seed. As the name suggests, the seed is a starting point for ongoing growth.

Rather than chasing the impossible goal of complete coverage, environment designers can initially underdesign the seed. That is, the designers do not create final solutions; they design spaces that knowledge workers can change and modify at use time.

The seeding phase requires system developers because the product is a complex software system. User participation is also necessary, because users have the knowledge necessary to decide what content the seed should include and how that content will need to evolve.

Although the SER model acknowledges that the initial seed cannot be complete, the seeding process still requires a substantial up-front investment. Existing software tools will likely have to be reimplemented or substantially adapted to function with information repositories. The repositories themselves must be designed to function with the tools (through underlying integrating mechanisms, such as critics). We have found that beginning with a community's existing information repositories and tools is effective. We then incrementally create prototypes that help developers and users understand how to cast their old information and technology into the new framework. This approach creates boundary objects for the users, letting them participate fully in the seeding.<sup>24</sup>

*Evolutionary growth.* This is the normal, operational phase of the SER model, in which the seed supports the three activities of the KM cycle. During this phase, the information repository plays two roles simultaneously: through dissemination it informs work, and through integration it accumulates the work products. Figure 5 depicts these roles as arrows.

A KM environment will experience several types of evolutionary growth, including

- *implicitly captured information* (for example, email and navigation traces).
- *explicitly produced information*, including finished work products (along with their rationale), which are collected in the catalog.
- *incremental formalizations*, representing information so that it can be connected conceptually and computationally to existing information in the repository. For example, a design rationale created during the project might be entered into the larger argumentative structure to show one alternative view or solution to a problem.

- *end-user modifications*, letting owners of problems and power users<sup>21</sup> extend the systems at the tool and at the content level.

An essential aspect of this phase is that the user community is responsible for changing the seed. Contributing domain knowledge should be part of everyone's job. But formalizing information and modifying system functionality might require significant programming knowledge. So, these tasks will be the responsibility of power users, who are technically inclined and motivated to do them.

The SER model assumes that some elements of an unself-conscious culture of design will emerge in the user community. Depending on this culture's strength, the evolutionary growth phase might last for an extended time period. However, as we discussed earlier, such decentralized evolution has its limits, and eventually the KM environment's usefulness and usability will suffer. When this happens, developers must come back into the picture to reseed the KM environment.

*Reseeding.* Reseeding is necessary for many reasons. For example, some incremental changes might point out fundamental limitations in the seed. Also, managing and combining many incremental changes might be difficult, and some incremental changes might make future changes more difficult. Reseeding is a complex process by which a group of users together with system developers take stock of the current system, synthesize its state, and reconceptualize it. This process produces a new system that can serve as the basis for future evolution. The evolution and reseeded cycle continues as long as people are using the system to solve problems.

Our experience with the SER model, as well as our observations of evolving software systems, indicates that periodic reseeded will be necessary, although the period between reseeded phases differs from community to community. It is necessary for two reasons. First, KM environments are embedded in a changing world and therefore must adapt. Small-scale modifications might suffice initially, but eventually any KM system will need to be modified in a way that is beyond even power users. Second, the contexts in which new knowledge is created are different from the contexts in which it will be reused. Restructuring this knowledge from its original form into a reusable form requires substantial effort.

## DynaSites

Developed at the University of Colorado, DynaSites (<http://seed.cs.colorado.edu/dynasites.documentation.fcgi>) is an environment for creating and evolving Web-based information repositories. It serves as a KM environment substrate (see Figure 5) to investigate KM processes in the context of the SER model. DynaSites currently houses 20 information spaces, all of which users can extend. It supports

- knowledge creation within the information spaces of individual projects,
- knowledge integration across the individual spaces by means of shared spaces, and
- knowledge dissemination by logically clustering related information.

As Figure 6 shows, the individual information spaces have four main components:

A threaded *discussion forum* belongs to a particular community. Dynasites currently has 16 discussion forums, four of which are active. The forums support a variety of communities, including university courses, research projects, and workshops. Anyone can create a discussion forum.

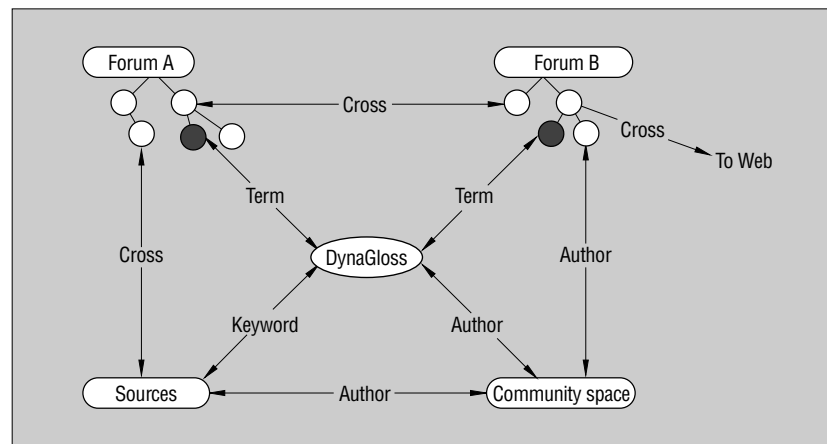
*Sources* is a shared repository for literature references, such as journal articles, conference proceedings, and Web sites. Each entry has a discussion thread that lets users hold open-ended discussions. Sources is open to all DynaSites users.

The *community space* holds *persona* pages for each DynaSites user. Users design per-

sonas, which contain information about the user, such as a picture, interests, a homepage URL, and whatever else the user wishes to share. Personas help users establish an identity within DynaSites and find others with whom to collaborate, based on mutual interests or complementary experiences. The community space currently contains 200 persona objects.

*DynaGloss* is a glossary of terminology open to all DynaSites users, who can annotate terms or redefine them when desired. DynaGloss currently contains 225 defined terms.

*Integration in DynaSites.* We use several strategies to link the information spaces in DynaSites (see Figure 6). Perhaps the most important are the *term links*, which enable DynaGloss to automatically integrate information across the entire DynaSites repository. For example, suppose the term "knowledge management" is defined in DynaGloss and appears in entries (shown cross-hatched in Figure 6) of both Forum A and Forum B. A user reading the entry in Forum A would see "knowledge management" represented as a link. Selecting the link would take her to the "knowledge management" entry in DynaGloss, which contains a definition and a list of all uses of the term throughout DynaSites. This list includes a link to the entry in Forum B containing "knowledge management." By following this link, the user would be likely to find a discussion relevant to Forum A, but possibly expressing a different perspective.



**Figure 6.** DynaSites provides several means to integrate the information repository. *Term links* bidirectionally connect the use of a term and its definition in DynaGloss. *Keyword links* connect records in sources with definitions in DynaGloss. *Author links* connect each contribution to the author's persona in the community space. Users create *cross links*, which connect arbitrary entries or connect an entry to any page on the Web.

Finally, she might follow the persona link for the entry in Forum B and become acquainted with a new collaborator.

The various linking strategies in Figure 6 create a rich web of information that connects ideas, people, and literature references. Because the system automatically creates and updates most of the links, information must be in a form that the system can interpret. For example, terms must have the same spellings as the glossary entries. The DynaSites information space's overall quality and integration requires effort and attention to detail that go beyond simply entering information. Without care, the information space can become unwieldy after a period of decentralized evolution. We are investigating how much extra effort users are willing to put into entering information, and what this effort's components are.

### Courses-as-seeds

This educational model attempts to explore the KM cycle in the context of university courses.<sup>25</sup> The goal is to establish a culture of collaborative knowledge creation that transcends the temporal boundaries of semester-based classes. In the spirit of the SER model, we conceptualize courses as seeds rather than finished products. Central to the courses-as-seeds model is an information repository that lets each course offering build on the products of prior semesters and serve as a forum for class discussions and a workspace for projects.

We now look at our initial attempt to implement this model. This implementation provided a concrete way to analyze our conceptual frameworks, such as the KM cycle and the SER model, as well as the supporting DynaSites technology.

The University of Colorado at Boulder is developing a major initiative called the *Alliance for Technology, Learning, and Society* ([www.colorado.edu/ATLAS](http://www.colorado.edu/ATLAS)). Part of the Atlas initiative is the *Technology Arts and Media* certificate program ([www.colorado.edu/ATLAS/certific.html](http://www.colorado.edu/ATLAS/certific.html)). In the context of the TAM program, we taught Designing the Information Society of the New Millennium ([www.cs.colorado.edu/~13d/courses/atlas-2000](http://www.cs.colorado.edu/~13d/courses/atlas-2000)) in the spring 2000 semester. (We will call this the TAM course.) This course's advertised goal was to let students explore how new media will affect learning, designing, and collaboration in the information society.

The class met twice a week; we based the activities on a series of assigned readings. We

assigned questions for each reading and asked students to post their responses in the discussion forum before the periods in which they discussed the responses. We strongly encouraged them to read and comment on each other's postings. Class discussions were based on the readings and responses but were not necessarily restricted to the reading topic. We assigned two projects in which students formed groups and selected their topics. The projects used a DynaSites forum for coordinating, communicating, and storing the project products.

At the semester's end, the forum contained 362 entries. Analysis of the information space indicates problems that limit the information's utility for future courses.<sup>25</sup> In terms

The design perspective assumes a culture in which management and workers see the workers as producers and managers of knowledge, rather than as consumers.

of the SER model, decentralized evolution over the semester resulted in an information space that required centralized integration.

The information's structure made sense to the creators but not to those who did not participate. During the course, the discussion threads were created to serve an unfolding discussion. As the discussions became focused, students articulated many nice insights. Users have difficulty finding these "nuggets" because they must read the entire thread (including branches). In effect, the nuggets are buried in the thread structure. Search mechanisms do not completely alleviate this problem, because a reader must know what to look for, and still must read the entries that the search returned.

The information produced during the course is also not well integrated in the larger DynaSites information space. Often, discussions relevant to terms defined in DynaGloss did not use exactly the same terminology. Therefore, the term-linking mechanism did not detect the discussions. In other cases,

forum entries mentioned literature references that might be helpful to future courses, but these references did not appear as Sources entries. So, they also became buried nuggets.

These situations are undesirable; they decrease the probability that students in the next TAM course will reuse the products. Students are unlikely to merely read them, let alone use them as building blocks, stable intermediate forms,<sup>26</sup> patterns,<sup>27</sup> or best practices<sup>15</sup> to develop the ideas further.

The reseeded process has involved editing the contents, formality, and structure of information spaces to make them more useful as building blocks for new knowledge. The DynaSites developers perform reseeded with TAM course participants, who own the information and therefore can best predict how it will be reused. The developers and participants collect and organize buried nuggets so that users can quickly find them. They edit selected entries so that the entries use terminology that the term-linking mechanism will pick up. Literature references are represented in Sources, where all DynaSites users can find and discuss them.

### Challenges

As we mentioned earlier, the design perspective assumes a culture in which management and workers see the workers as producers and managers of knowledge, rather than as consumers. In this culture, workers are motivated to share their knowledge rather than hoard it as "job security." Achieving this culture, however, involves major challenges.

### Creating new mind-sets and KM cultures

Our KM perspective requires a cultural transformation in which all stakeholders must learn new relationships between practices and attitudes. Our initial steps have been to self-apply our theories and technologies in our university context through the courses-as-seeds model. This context is convenient because courses provide access to communities in which the risk of trying new practices is acceptable (and even educational) and because stakeholders will be more forgiving of immature technologies.

*Education reform.* What is more important, we feel that the traditional educational model, like traditional KM models, needs serious reform.

The courses-as-seeds model's premise is that the traditional education paradigm is inappropriate for studying the types of open-

ended and multidisciplinary problems that are most pressing to our society. These problems, which typically involve a combination of social and technological issues, require a new paradigm of education and learning skills, including self-directed learning, active collaboration, and consideration of multiple perspectives. Problems of this nature do not have “right” answers, and the knowledge to understand and resolve them is changing rapidly, requiring an ongoing and evolutionary approach to learning.

The courses-as-seeds model represents a system of values, attitudes, and behaviors that is radically different from the traditional educational culture, which views courses as finished products and students as consumers. Courses-as-seeds aims to create a culture based on a “designer mind-set” that emphasizes habits and tools that empower students to actively contribute to the design of their education (and eventually to the design of their lives and communities).

*Beyond consumers.* Evaluation of our courses shows the difficulties of changing the mind-sets that students have been taught over years in the educational system and that continue in the workplace. The collaboration and evolutionary growth that the SER model postulates is impossible in communities where most members regard themselves as consumers.<sup>16</sup> Individuals must have the opportunity to evolve into power users and codevelopers who use and can, at the same time, modify and extend their KM environments if necessary. Toward that end, information technology can help us understand and exploit software’s malleability, which will let us construct knowledge collaboratively in the context of work.

Arguing for users being designers and not just consumers requires a deep understanding of delegation in a society characterized by a division of labor (see Table 4, row 4). Delegation is desirable when the delegator does not possess the knowledge or skill to accomplish a task directly and when the task can be specified in enough detail to be entrusted to someone else. Professional expertise has its place—if it is used properly. For example, professional developers are necessary during the SER model’s seeding and reseeding phases, because they possess the technical skills necessary to substantially modify the KM environment.

### Motivation

An important nontechnical challenge for

collaborative construction and evolution of information repositories is to take motivation seriously. Our experiences with courses-as-seeds illustrates this challenge.

In the courses-as-seeds model, the instructors intended to spur peer-to-peer interaction by assigning reading materials and requiring students to post their responses in the forum. The instructors reasoned that because student’s postings would be available to their peers, interesting discussion based on these postings would follow. Because instructors assumed that students would be intrinsically motivated to interact with their peers, they did not make this an explicit part of the grading criteria.

The instructors’ assumption did not hold. The reading assignments dominated activity

Sustained collaborative work practices require an incentive to create social capital by rewarding stakeholders for contributing and receiving knowledge as a member of a community.

in the forum; students posted long responses but only extremely rarely commented on another student’s response. The high participation rates and considerable length of the assigned postings show that students were motivated to spend considerable time and effort fulfilling the explicit requirements for a good grade. But they were not motivated to spend the additional time required to read and comment on the responses of their peers.

The course design did not consider carefully enough the competing demands from other classes for the student’s time and attention. In effect, the course design sent students a mixed message. The graded assignment policy reinforced the traditional model to which the students were accustomed, and might have led them to do only what they considered necessary for a good grade. On the other hand, the course’s content and the rhetoric of the instructors implied a different model that assumed students would be motivated to go beyond the minimum.

Sustained collaborative work practices

require an incentive to create *social capital*<sup>28</sup> by rewarding stakeholders for contributing and receiving knowledge as a member of a community. Social capital is based on these concepts:

- Human beings have an innate drive to compete for social status.
- What you give away, not what you control, determines social status.
- Prestige is a good way to attract attention and cooperation.
- Utilization is the sincerest form of flattery.

The Experts Exchange ([www.experts-exchange.com](http://www.experts-exchange.com)) is an example of a *gift culture* that provides social capital. It is a vast, evolving repository of answers to a wide variety of technical questions. Users compete for *expert points* by giving good answers to questions from other users. Users amassing the highest number of expert points receive recognition from the community and are listed in the “winner’s circle” for all to see.

Users can spend *question points* to ask questions or see previously answered questions. When users become a member of Experts Exchange, they receive enough points to see the answers to approximately 15 questions. However, if they wish to retain their privileges for a sustained time, they must earn more through various activities.

As you can see, Experts Exchange is a knowledge-sharing culture built on a mutually beneficial relationship: questioners receive answers, and experts gain social capital.

The paradise of shared knowledge isn’t just happening. Knowledge isn’t shared because management does not want to share authority and power.  
—Shoshana Zuboff

**T**echnology alone will not solve the difficult problems of KM.<sup>1,15</sup> Knowing is a human act. Although new technologies are important and necessary for progress in KM, they are insufficient.

KM forces us to transcend individual perspectives. Until recently, computational environments focused on the needs of individual users. As more people use computers for more complex tasks, we are realizing that we need environments supporting social interactions among communities of practice and

## The Authors



**Gerhard Fischer** is a professor of computer science, a fellow of the Institute of Cognitive Science, and the director of the Center for Lifelong Learning and Design (L3D) at the University of Colorado at Boulder. His research includes new conceptual frameworks and new media for learning, working, and collaboration; human-computer interaction; cognitive science; artificial intelligence; (software) design; and domain-oriented design environments. Contact him at the Center for LifeLong Learning and Design, Dept. of Computer Science, Univ. of Colorado, Campus Box 430, Boulder, CO 80309-0430; gerhard@cs.colorado.edu; www.cs.colorado.edu/~gerhard.



**Jonathan Ostwald** is a research fellow at the Center for Lifelong Learning and Design (L3D) at the University of Colorado at Boulder. His research interests include human-computer interaction; computer support for design, learning, and collaboration; and evolutionary and participatory models of software development. He received his PhD from the University of Colorado in Boulder. Contact him at the Center for LifeLong Learning and Design, Dept. of Computer Science, Univ. of Colorado, Campus Box 430, Boulder, CO 80309-0430; ostwald@cs.colorado.edu; www.cs.colorado.edu/~ostwald.

communities of interest. However, this perspective does not necessitate the development of environments in which the group's interests inevitably supersede the individual's. Individuality makes a difference, and communities get their strength to a large extent from the creativity and engagement of the individual. An important challenge will be to gain a better understanding of the relationship between the individual and the community.

Ongoing collaborative knowledge construction and sharing (in the context of creative design activities) are difficult processes. To make real progress with KM requires changing work practices, mind-sets, and reward structures. A student participating in our course characterized the ultimate challenge to KM: "Collaborative systems will not work in a noncollaborative society." ■

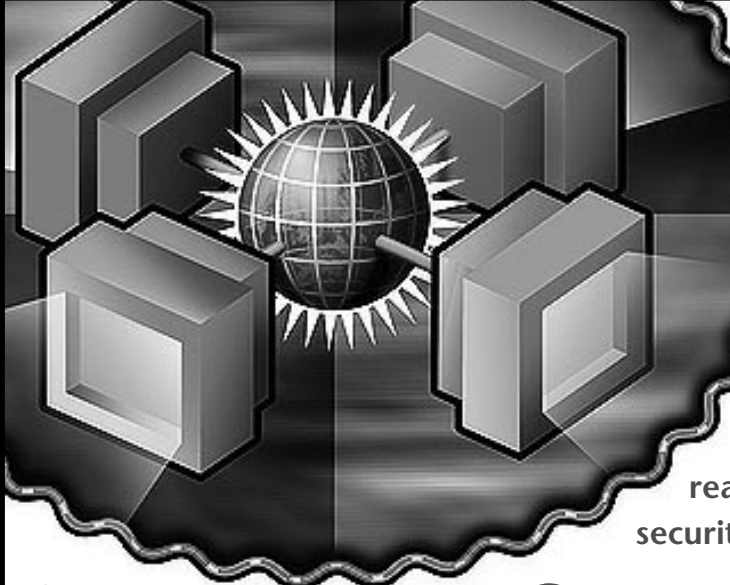
### Acknowledgments

We thank the members of the University of Colorado's Center for LifeLong Learning and Design, who have made major contributions to the conceptual frameworks and systems described in this article. National Science Foundation Grants REC-9631396 and IRI-9711951; Office of Naval Research Cooperative Agreement N66001-00-1-8964; Software Research Associates, Tokyo; PFU, Tokyo; and the Coleman Foundation all supported this research.

### References

1. J.S. Brown and P. Duguid, *The Social Life of Information*, Harvard Business School Press, Boston, 2000.
2. D.A. Schön, *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, New York, 1983.
3. F. Shipman and R. McCall, "Supporting Knowledge-Base Evolution with Incremental Formalization," *Human Factors in Computing Systems, INTERCHI '94 Conf. Proc.*, ACM Press, New York, 1994, pp. 285-291.
4. E.G. Arias et al., "Transcending the Individual Human Mind: Creating Shared Understanding through Collaborative Design," *ACM Trans. Computer-Human Interaction*, vol. 7, no. 1, Mar. 2000, pp. 84-113.
5. G. Fischer, "Domain-Oriented Design Environments," *Automated Software Eng.*, vol. 1, no. 2, June 1994, pp. 177-203.
6. E. Wenger, *Communities of Practice: Learning, Meaning, and Identity*, Cambridge Univ. Press, Cambridge, UK, 1998.
7. J. Bruner, *The Culture of Education*, Harvard Univ. Press, Cambridge, Mass., 1996.

8. J. Greenbaum and M. Kyng, eds., *Design at Work: Cooperative Design of Computer Systems*, Lawrence Erlbaum Associates, Hillsdale, N.J., 1991.
9. A. Dieberger et al., "Social Navigation: Techniques for Building More Usable Systems," *Interactions*, vol. 7, no. 6, Dec. 2000, pp. 36-45.
10. W.C. Hill et al., "Edit Wear and Read Wear," *Proc. CHI '92 Conf. Human Factors in Computing Systems*, ACM Press, New York, 1992, pp. 3-9; www.acm.org/pubs/articles/proceedings/chi/142750/p3-hill/p3-hill.pdf. (current 14 Feb. 2001).
11. H.E. Pashler, *The Psychology of Attention*, MIT Press, Cambridge, Mass., 1998.
12. T.K. Landauer and S.T. Dumais, "A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge," *Psychological Rev.*, vol. 104, no. 2, Feb. 1997, pp. 211-240.
13. J. Orr, *Talking about Machines: An Ethnography of a Modern Job*, ILR Press/Cornell Univ. Press, Ithaca, N.Y., 1996.
14. J. Grudin, "Groupware and Social Dynamics: Eight Challenges for Developers," *Comm. ACM*, vol. 37, no. 1, Jan. 1994, pp. 92-105.
15. R.G. Smith and A. Farquhar, "The Road Ahead for Knowledge Management: An AI Perspective," *AI Magazine*, vol. 21, no. 4, Winter 2000, pp. 17-40.
16. G. Fischer, "Social Creativity, Symmetry of Ignorance and Meta-Design," *Knowledge-Based Systems J.*, vol. 13, nos. 7-8, Dec. 2000, pp. 527-537.
17. L.G. Terveen, P.G. Selfridge, and M.D. Long, "Living Design Memory: Framework, Implementation, Lessons Learned," *Human-Computer Interaction*, vol. 10, no. 1, 1995, pp. 1-37.
18. J. Thomas, *Welcome to Information about Knowledge Management*, 2001, www.truthtable.com/know.html. (current 26 Feb. 2001).
19. T. O'Reilly, "Lessons from Open Source Software Development," *Comm. ACM*, vol. 42, no. 4, Apr. 1999, pp. 33-37.
20. E.S. Raymond, *The Cathedral and the Bazaar*, 1998, www.tuxedo.org/~esr/writings/homesteading/cathedral-bazaar (current 15 Feb. 2001).
21. B.A. Nardi, *A Small Matter of Programming*, MIT Press, Cambridge, Mass., 1993.
22. C. Alexander, *Notes on the Synthesis of Form*, Harvard Univ. Press, Cambridge, Mass., 1964.
23. G. Fischer et al., "Embedding Critics in Design Environments," *Readings in Intelligent User Interfaces*, M.T. Maybury and W. Wahlster, eds., Morgan Kaufmann, San Francisco, 1998, pp. 537-561.
24. J. Ostwald, *Knowledge Construction in Software Development: The Evolving Artifact Approach*, Ph.D. dissertation, Dept. of Computer Science, Univ. of Colorado, Boulder, Colo., 1996; www.cs.colorado.edu/~ostwald/thesis (current 15 Feb. 2001).
25. R. dePaula, G. Fischer, and J. Ostwald, "Courses as Seeds: Expectations and Realities," to be published in *Proc. Euro-CSCL 2001*, Maastricht, Netherlands, 2001.
26. H.A. Simon, *The Sciences of the Artificial*, 3rd ed., MIT Press, Cambridge, Mass., 1996.
27. C. Alexander et al., *A Pattern Language: Towns, Buildings, Construction*, Oxford Univ. Press, New York, 1977.
28. E.S. Raymond, *Homesteading the Noosphere*, 2000, www.tuxedo.org/~esr/writings/homesteading/homesteading (current 15 Feb. 2001).



cluster computing  
 collaborative computing  
 distributed agents  
 distributed databases  
 distributed multimedia  
 grid computing  
 middleware  
 mobile & wireless  
 operating systems  
 real-time systems  
 security

# Distributed Systems Online

computer.org/dsonline

## Intelligent Systems

### Advertiser/Product Index January/February 2001

	Page No.	Advertising Sales Offices
<b>CHI 2001</b>	21	<b>Sandy Aijala</b> 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314; phone +1 714 821 8380; fax +1 714 821 4010; saijala@computer.org.
<i>IEEE Distributed Systems Online</i>	73	
<i>IEEE Intelligent Systems</i>	14, back cover	<b>Advertising Contact:</b> Debbie Sims, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314; phone +1 714 821 8380; fax +1 714 821 4010; dsims@computer.org.
<i>IEEE Internet Computing</i>	58	

Boldface denotes advertisers in this issue.

For production information, and conference and classified advertising, contact Debbie Sims, *IEEE Intelligent Systems*, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314; phone (714) 821-8380; fax (714) 821-4010; dsims@computer.org; http://computer.org.