Contribution to a Book (eds: Marleen Huysman and Volker Wulf) about ***"Social Capital"*** (based on the May 2002 Workshop, Amsterdam)

Fostering Social Creativity by Increasing Social Capital

Gerhard Fischer[1], Eric Scharff [1],Yunwen Ye[1,2]

[1] Center for LifeLong Learning and Design
Department of Computer Science
University of Colorado
Boulder, CO 80303-0430, USA
+1-303-492-1592
{gerhard,scharffe,yunwen}@cs.colorado.edu

[2]SRA Key Technology Laboratory, Inc.
3-12 Yotsuya, Shinjuku, Tokyo 160-004, Japan
+81-3-3357-9011

Abstract

Complex design problems require more knowledge than any single person can possess, and the knowledge relevant to a problem is often distributed among all stakeholders who have different perspectives and background knowledge, thus providing the foundation for *social creativity*. Bringing together different points of view and trying to create a shared understanding among all stakeholders can lead to new insights, new ideas, and new artifacts. Social creativity can be supported by innovative computer systems that allow all stakeholders to contribute to framing and solving these problems collaboratively.

Technology alone, however, is not the complete answer to social creativity. *Social capital* that characterizes the features of a social group—such as networks, norms, and trust, which all facilitate coordination and cooperation for mutual benefit—is of critical importance to foster social creativity.

In this paper, we discuss (1) the roles that social capital plays in facilitating social creativity and (2) approaches to increase social capital. We start by analyzing existing success models (Open Source and Experts Exchange) that support collaborative knowledge construction in order to create a conceptual framework to understand the social-technical aspects of promoting social capital. We further illustrate this conceptual framework with our own efforts in creating *social capital-sensitive computer systems* (e.g., Evolutionary Reuse Repositories, Envisionment and Discovery Collaboratory, and Courses-as-Seeds) that support collaborative design, problem solving, and knowledge construction. These systems show the importance of encouraging users to act as active contributors and illustrate some of the motivational challenges upon which these systems rely.

The *assessment* of these activities provides evidence that collaborative technologies are necessary, but not sufficient, to create more collaborative communities. Without a deep understanding of the motivation, reward structures, and the creation of new mindsets and organizations based on a greater emphasis of social capital, the impact of these new technologies will be negligible.

Keywords

social creativity; social capital; social capital-sensitive systems; meta-design; seeding, evolutionary growth, reseeding (SER) model; Open Source; Envisionment and Discovery Collaboratory; reusable software repositories; motivation; trust; self-application

Table of Contents

List of Figures

List of Tables

1 Introduction

Complex design problems require more knowledge than any single person can possess, and the knowledge relevant to a problem is often distributed among all stakeholders who have different perspectives and background knowledge, thus providing the foundation for *social creativity* [Fischer 2000]. Bringing together different points of view and trying to create a shared understanding among all stakeholders can lead to new insights, new ideas, and new artifacts. Social creativity can be supported by innovative computer systems that allow all stakeholders to contribute to framing and solving these problems collaboratively. Such systems need to be designed from a *meta-design* perspective by creating environments in which stakeholders can act as active contributors and be more than just consumers [Fischer 1998].

To foster social creativity, people need to be motivated to be active contributors, for example. to add content to organizational memories, to share tools that they have developed, to bring their unique knowledge into a discussion, or to formulate their ideas in a way that other stakeholders can understand. We are particularly interested in socio-technical environments in which these activities are not performed as a work assignment (i.e., they are not required nor are they explicitly rewarded), but in which people are motivated to accumulate *social capital;* the incentive is to be a good colleague, to contribute and receive information as a member of the community, to acquire a reputation of being a good citizen, and to jointly construct artifacts that could not be developed individually.

2 Social Creativity and Social Capital: A Conceptual Framework

The Western belief in individualism romanticizes the perception of a solitary process in creative activities, but the reality is that scientific and artistic forms emerge from the joint thinking, passionate conversations, and shared struggles common in meaningful relationships [John-Steiner 2000]. The power of the unaided individual mind is highly overrated [Salomon 1993]. Although creative individuals are often thought of as working in isolation, intelligent and creative results come mainly from interaction and collaboration with other individuals. Creative activity 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. Much human creativity arises from activities that take place in a social context in which interaction with other people and the artifacts that embody group knowledge are important contributors to the process. Creativity [Florida 2002] does not happen solely inside a person’s head, but in the interaction between a person's thoughts and a socio-cultural context [Engeström 2001]. Situations that support social creativity need to be sufficiently open-ended and complex that users will encounter breakdowns [Schön 1983]. As any professional designer knows, breakdowns—although at times costly and painful—offer unique opportunities for reflection and learning.

The concept of *social capital* provides an anchor to analyze the non-technical and non-managerial aspects of social creativity. Social capital creates the resources upon which a person can draw to obtain knowledge, cooperation, and help from others.

2.1 Social Creativity

To make *social creativity* [Fischer 2000] a reality, we have explored new forms of knowledge creation, externalization, integration, and dissemination based on the observation that the scarce resource in the information age is not information—rather, it is the human resource to attend to this information. One aspect of supporting social creativity is the externalization of an individual’s and a group’s tacit knowledge [Polanyi 1966]. Individual tacit knowledge means intuition, judgment, and common sense: the capacity to do something without necessarily being able to explain it. Group tacit knowledge means knowledge existing in the distinct practices and relationships that emerge from working together over time. *Externalizations* [Bruner 1996] support social creativity in the following ways:

- they cause us to move from vague mental conceptualizations of an idea to a more concrete representation of it;
- they provide a means for others to interact with, react to, negotiate around, and build upon an idea;
- they allow more voices from other stakeholders to be brought in; and
- they create a common language of understanding (including boundary objects that are understandable across different domains [Arias and Fischer 2000]).

Creating these externalizations requires active contributors and not just consumers. Externalizations of individual knowledge make it possible to accumulate the knowledge held by a group or community. An important challenge for social creativity is to capture a significant portion of the knowledge generated by work done within a community. Experiences with organizational memories and collaborative work have exposed two barriers to capturing information. First, individuals must perceive a value in contributing to an organizational memory that is large enough to outweigh the effort [Grudin 1994]. Second, the effort required to contribute to organizational memory must be minimal so it will not interfere with performing the work at hand [Carroll and Rosson 1987]. Since human beings often try to maximize utility in the decision-making process [Reisberg 1997], increasing the value and decreasing the effort of knowledge externalization and sharing are essential. No objective evaluation of value and effort exists: the evaluation is subject to each person's perception and background.

Many factors play a role in the evaluation of the perceived value; among these are the three major dimensions of economical, intellectual, and social values. Economical capital includes the monetary rewards that a person can receive; intellectual capital refers to the intellectual satisfaction that a person derives from making a contribution, such as the boost of confidence, the self-assurance of individual capability, and the acquisition of knowledge and skills,; social capital refers to the influence brought about by individuals' behaviors on the relations with their social-cultural environment.

Social creativity entails taking a new perspective on how we design the supporting technological, social, and organizational environments. Without this perspective, technology to support working and learning is often designed in ways that fail to support social creativity. Organizational aspects (e.g., course structures and curricula in educational settings) are often concerned more with transmitting facts and basic skills and do not adapt well to open-ended problem solving and collaborative learning.

An important prerequisite to bring social creativity alive is that media and environments be available to support *meta-design* [Fischer and Scharff 2000]. The perspective of meta-design characterizes objectives, techniques, and processes that allow users to act as designers and be creative [Henderson and Kyng 1991]. The need for meta-design is founded on the observation that design in the real world requires open systems that users can tailor, customize, modify, and evolve [Fischer and Girgensohn 1990; Mackay 1990; MacLean, Carter et al. 1990; Oppermann 1994]. Because problems cannot be completely anticipated at design time (when the system is developed), users at use time will discover mismatches between their problems and the support that a system provides.

2.2 Social Capital

Social creativity is essential to support collaborative design through the active participation of all stakeholders. Although previous research in knowledge management, organizational memory, and groupware systems [Preece 2002] has recognized the important role that social context plays (such as motivation and trust), a unified conceptual framework does not exist to analyze such a role. *Social capital* characterizes the interpersonal relationships that an individual has with other members in a surrounding community, and it provides the basis for analyzing the sense of community and the degree to which the individual is connected with others in the community.

The concept of social capital first appeared in the writing of Lyda Judson Hanifan [Hanifan 1916], who used it to describe those tangible substances, such as good will, fellowship, sympathy, and social interactions, that "count most in the daily life of the people" living in rural communities. Recently, this concept has been widely applied to a variety of disciplines, including political science [Putnam 1995], sociology [Coleman 1988], education [Bourdieu 1983], business management [Meyerson 1994], organizational theory [Cohen and Prusak 2001], and collaborative software construction [Raymond 1999]. Many definitions of social capital exist, including:

- *"features of social organization, such as networks, norms, and trust, that facilitate coordination and cooperation for mutual benefit."* [Putnam 1995]
- *"an instantiated informal norm that promotes cooperation between two or more individuals."* [Fukuyama 1999]

A useful framework for understanding the role of social capital in fostering social creativity as well as approaches to increasing social capital is proposed by Nahapiet and Ghoshal [Nahapiet and Ghoshal 1998], who define social capital as *"the sum of the actual and potential resources embedded within, available through, and derived from the network of relationships possessed by an individual or social unit."* They identify the three operational dimensions in social capital:

- *the structural dimension:* a relationship network that connects people and helps individuals to find people for assistance or cooperation;
- *the relational dimension:* the sense of trust that individuals have toward each other along connections; and
- *the cognitive dimension:* the bonding force, such as shared understanding, interest, or problems, that holds the group together.

This definition indicates that social capital exists at both the individual level and the group level (social unit). Most studies on social capital have focused on the group level, namely, the active connections among the group members: the trust, mutual understanding, and shared values and behaviors generating the following benefits for the group:

- better knowledge sharing and transfer due to established trust, shared language and goals, and informal ties; and
- lower transaction costs due to trust and cooperation make less necessary any costly formal coordination mechanisms such as contracts and the like.

The social capital of a group depends on the social capital held by its individual members, although without a group that provides the social context, an individual could not have any social capital. Within a given social context, an individual's social capital is the actual and potential resources that the individual could draw on for acquiring cooperation from other members.

In systems that require user participation and collaboration, social capital, by its definition, is an important factor to consider. We use the term *social capital-sensitive system* to refer to systems whose success is influenced by social capital and that are designed to take issues of social capital, explicitly or implicitly, into consideration.

2.3 The Seeding, Evolutionary Growth, and Reseeding Process Model

The development and deployment of computer systems that support social creativity require the continual participation and contribution of developers and users not only at the *design time* (when the original system is developed) but also at the *use time* (when people use and evolve the systems in their activities). System development techniques have typically emphasized a strong distinction between the activities of designing a software system and the use of that system. Even efforts such as participatory design [Schuler and Namioka 1993], which has improved user involvement in software design, have emphasized system development and use separations. In order to accommodate open problems and continuously changing requirements, it is necessary to avoid separation of design and use and to create models in which continuous change can take place.

The *Seeding, Evolutionary Growth, and Reseeding (SER)* model is a process model that aids in the understanding and development of systems that evolve over time [Fischer, Grudin et al. 2001]. This model has been developed based on experience with analyzing and building systems supporting collaborative design. The SER model helps in understanding social creativity by identifying important phases in the evolution of systems supporting collaborative construction and evolution. Each of the phases of the SER model raises questions that systems supporting social creativity must face.

Social creativity in the SER model is a continuous process that encourages users to act as designers. For a system that relies on participation of a community to succeed, the social and technical aspects of the system must both be designed to leverage and increase the social capital of a community. We call such systems *social-capital sensitive systems* because their successful operation is strongly affected by the social capital in the environment where they are deployed

The SER model postulates three phases of development. The *seeding* process involves the creation of an initial system that can grow over time. During *evolutionary growth*, users make changes to and extend the initial seed through their own work. When the incremental changes in evolutionary growth are no longer practical, a *reseeding* phase takes place in which the incremental changes are organized, reformulated, generalized, and incorporated into the initial seed. Cycles of evolution and reseeding continue as long as the community is actively developing and using the software.

The first goal of providing a seed is to provide immediate benefits to users to attract them to the system. The second equally important goal of seeding is to engage in meta-design by creating environments that allow users to become active designers through extending, refining, and augmenting the existing system. An initial system *open to user participation* is the prerequisite to support social creativity in the context of collaborative knowledge construction and design.

For the *seeding phase*, the following questions relevant to a social capital perspective are important:

- How can a seed be perceived as providing increased value to members of a community so that they will feel motivated to participate?
- What aspects of a seed can reduce the effort needed to use and modify that seed?
- Who must participate in the creation of the seed?
- How does the seed balance the goals of the initial developers and the needs of the community at large?

During the *evolutionary growth phase*, the users must incrementally change a system in the context of their work, generating the following questions and challenges for *social capital-sensitive systems:*

- What kinds of extension mechanisms are necessary?
- What motivates people to contribute?
- Is there a collaborative process that encourages people to make and share contributions?
- Do participants receive individual benefits and social rewards that increase the utility and justify the effort of contribution?

Managing incremental changes is difficult, and the reorganization of the reseeding phase is an attempt to synthesize these incremental changes and create a new stable system upon which new changes can be created. Reseeding requires major modifications and it raises the following issues:

- When is reseeding necessary?
- If reseeding requires a great deal of effort, who perceives the value in doing it?
- How can reseeding efforts integrate the modifications of members of the community and continue to acknowledge the importance of individual contributors?

The SER model characterizes the typical life cycle of computer systems that support social creativity in the context of collaborative design. The success of such systems requires social capital that motivates users to actively participate and facilitates collaboration and coordination.

3 Analysis of Existing Success Models

In order to develop a conceptual framework for social capital-sensitive systems, we have analyzed (1) *open source* developments (a methodology to engage the “talent pool of the whole world”), and (2) the Experts Exchange (an environment in which information technology experts compete and collaborate with each other to provide specific solutions to specific problems and in which the experts receive recognition by the questioners). These analyses were based on numerous sources, including first-hand participation in open source and Experts Exchange efforts, long-term evaluation of the communication and evolving artifact in open source communities, and comparisons with other ongoing open source analyses [Scharff 2002].

3.1 Open Source

Open source development [Raymond and Young 2001] is an activity in which a community of software developers collaboratively constructs systems to help solve problems of shared interest and for mutual benefit. Contemporary perspectives on open source tend to emphasize the technical aspects of projects, the projects themselves [DiBona, Ockman et al. 1999; Mockus, Fielding et al. 2000], the licenses under which the software is shared [Perens 1999], or the use of software in a particular domain [Davis, O'Donovan et al. 2000; Claypool, Finkel et al. 2001]. Although these are important aspects of open source software, they do not acknowledge the collaborative processes involved in producing the software. Understanding open source as a process of *collaborative design and construction* highlights the socio-technical aspects of open source software as *social capital-sensitive systems* in which the participation of a community is vital to its success, and the final product emerges from the contributions of the entire community. Development in the open source community has been characterized by principles related to social capital, such as *"(1) in gift cultures, social status is determined not by what you control but by what you give away, (2) prestige is a good way to attract attention and cooperation from others, and (3) utilization is the sincerest form of flattery* [Raymond and Young 2001].*"*

Open source provides evidence of social capital both as a *noun* (the resources that a community has that facilitate group interaction and sustainability) and as a *verb* (the processes that sustain ongoing community activity). Seen from a collaborative design perspective, the ingredients that make up an open source community include a group of participants willing to contribute to the community, a concrete software artifact to which explicit extensions are made, and a set of collaborative technologies that help the community communicate and coordinate the objects that they are producing.

The collaborative processes used to coordinate activity in the project form the basis of joint activity. A defining characteristic of open source software is that the source code for that software can be obtained and modified by anyone who wishes to do so. The ability to change source code is an enabling condition for collaborative construction of software because it allows software developers to make changes to the behavior of the software. This simple condition changes software from a fixed entity that is produced and controlled by a closed group of designers to an open effort that allows a community to collaboratively design based on personal desires following the framework provided by the seeding, evolutionary growth, and reseeding process model. Open source invites passive consumers to become active contributors. Of course, modifying software requires the time, inclination, and technical aptitude to do so—conditions that do not exist in all communities.

Studies and analyses of open source software [Scharff 2002] highlight the importance of the three dimensions of social capital (see Section 2.2):

1. The structural dimension:

- *Coordinating distributed work.* Collaborative technology helps to provide the structure through which collaboration in the community can take place. Open source projects rely heavily on collaborative technology to coordinate the software itself (using technical tools such as source code management systems [Fogel 2000]) as well as to communicate about the project (using communication tools such as chat, email, and threaded discussions). Asynchronous tools are favored due to large geographically distributed communities, and face-to-face meetings are rare. Therefore, maintaining social cohesion requires the active use of collaborative technology. Collaborative technology helps to create a sense of community and supports work. Different communities use technology differently, but the need for focused technical discussion, general discussion, and off-topic social discussion is commonplace.
- *Role distribution.* The social capital in an open source community is not based on a set of homogenous individuals who could merely be replaced by other competent participants. Rather, the capital is based on the complementary skills and interests of participants. In a healthy open

source project, the skill sets of different participants will complement each other [Campbell 1969]. Different participants take on different roles that facilitate the rapid change and creation of stable releases, including testing, contributing new changes, coordinating releases, and maintaining documentation. An important aspect of the collaborative approach is to help individuals find tasks in which their talents would be best applied.

2. The relational dimension:

- *Supportive process.* The process of adopting changes proposed by the community creates interesting conflicts and trade-offs. On the one hand, people must be given the feeling that their contributions are worthwhile and that they will benefit the group; this facilitates ongoing participation. On the other hand, not all changes are universally useful, and conflicting extensions must somehow be mitigated. The most visible part of this process manifests itself as a single project leader who has the ultimate (dictatorial) control over what should be included in the system. However, the change adoption process (as an essential component of reseeding) is far more complex and relies heavily on discussion and comparison of changes in online discussions. Even if changes are ultimately not adopted, community members tend to be actively supportive of all contributions, trying to make contributors realize that their input is worthwhile. When this process breaks down, open source projects tend to split into separate projects and bifurcate the community, although this situation is extremely rare [Raymond 1998].

3. The cognitive dimension:

- *Peer pressure.* Public code deliverables encourage developers to write good code. They take pride in their contribution, and low-quality contributions are detrimental because it lowers the contributors' reputation.
- *Motivation.* Similar to community participation in other areas, participation is a mix of intrinsic and extrinsic motivation. As in the relational dimension of social capital, open source projects rely on a group of motivated and mutually trusting participants. Without active participation, open source projects would simply be static pieces of software rather than ongoing community activities. In successful open source projects, motivation provides a positive feedback loop by which a group of core participants excite and motivate other designers to join a project. Although open source participants typically do not get paid for their contributions, other forms of external compensation, such as gaining a reputation among peers governed by what they contribute to the community [Raymond and Young 2001], exist. This extrinsic motivation is complemented by the cognitive dimension of social capital, in which members participate in activities that they find intellectually stimulating and personally enriching. Participating in an open source project is a way to learn about a new technology that may be useful to further professional development of technically inclined participants. It can also be an investment in the future, perhaps by gaining socio-economic status through a better job, or through meeting people leading to future enhanced professional relationships.

3.2 Experts Exchange

The ***Experts Exchange*** ([Experts-Exchange 2002]; see Figure 1) is a Web-based knowledge-sharing environment that exemplifies many of the important aspects of social creativity and social capital. Thousands of experts regularly participate in answering myriad questions posed by the community. Unlike some systems that separate novices and experts, anyone can pose a question or become an expert. The unique characteristic of the Experts Exchange is a point system to motivate active participation. When users join, they are given a number of points. They can offer points to others in exchange for answering questions, and can gain points by answering questions posed by other users. Questions are organized into categories related to topics important in information technology, such as Linux and Java programming. Persons asking questions have the ability to judge the quality of answers posed to their questions. Given this system of demands, recognition, and rewards, the Experts Exchange has developed an active user community of questioners and answerers.

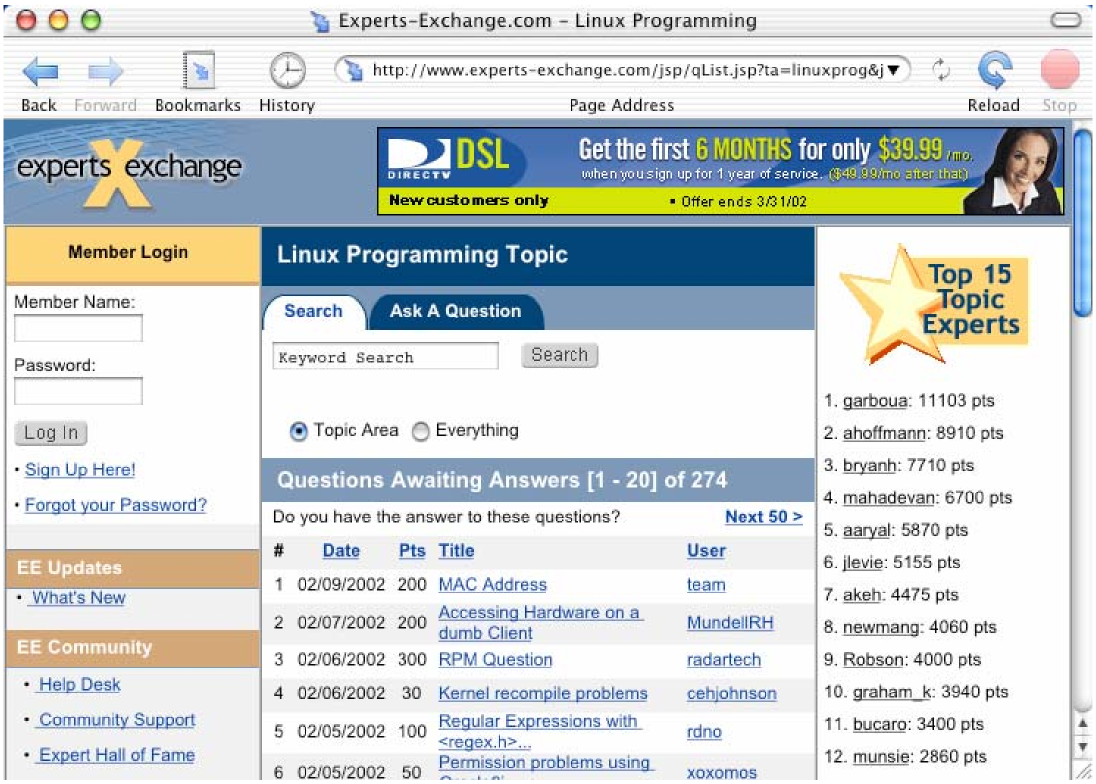

Figure 1: A Screen Image of the Expert Exchange

Examining the Experts Exchange as a social capital sensitive system using the SER model illustrates the importance of social creativity and social capital at both the individual and group levels. The *seed* for the Experts Exchange is a Web site that allows people to post questions, respond to already posted questions, and evaluate these responses. Incentives include giving new users initial points (so that they can ask questions). The effort in posting new questions is low, and the potential value is high if there are experts who can and will answer the question. The system evolves through the addition of new questions and answers to posed questions. When someone submits a candidate answer to a question, the user who posted the question can evaluate the quality of the answer and determine if the points should be awarded for that answer. The Experts Exchange site itself has evolved, including: (1) a redesign of the user interface and the addition of new features such as finding jobs for users who demonstrate a certain expertise; (2) a reseeding of the content by placing previously answered questions and answers into the "IT Knowledge Base" (providing a collection of questions and answers available as a service for commercial subscription); and (3) support for linking questioners with potential experts, providing subject-based categories for questions, and notifying people when new questions are posted. The reorganization and repackaging of information captured in questions and answers has been a primary business model for the Experts Exchange.

The Experts Exchange has relied on a mix of external rewards and internal motivation. The point system has served as a metric for the activity and aptitude of a member in a community. Over time, some users have amassed large numbers of points due to the quality of their answers, highlighting the important individual and social aspects of creating social capital, including the following:

- **Explicit recognition and reputation enhancement.** Points provide a mechanism not only to reward individuals for participating, but to provide explicit public recognition of an individual's contribution to the community. The top users with the most points in a given forum are presented in a list prominently displayed upon entering the forum. Thus, if someone wishes to demonstrate her or his skill (such as being a Java programming expert), she or he can answer questions posed in the Java programming forum. This has encouraged informal competition for recognition to be the expert in a given area. Public display of points and recognition for valuable contributions gives contributors an easily quantified form of satisfaction.

- **Community awareness.** Due to the large number of questioners, answerers, and posed questions, it is difficult to get a sense of the activity of members in a community. Unlike a threaded discussion or ongoing coordinated activity, users will usually not be aware of work outside of the relatively small numbers of questions they have asked or answered. The public display of members' points is an indicator of the activity of members of the community, making users aware of contributors who are currently involved in answering questions and people who have been involved in the past [Erickson and Kellogg 2001].
- **Positive reinforcement.** The point system emphasizes the positive aspects of contributing valuable answers while minimizing potentially socially detrimental behavior. Good answers are publicly praised and rewarded, but weak answers are not presented for public ridicule, and users are not criticized for the questions they were not able to answer well. This creates an environment that encourages a competition to succeed and assist others without emphasizing a "zero-sum game" in which winning, by definition, involves losing.
- **Encouragement of information exchange.** Points give potential contributors an explicit reason to contribute, based on the personal benefits described above. *The points contributors have are one indication of their social capital in the community.* They can use the points to obtain help from others by asking questions, thus leading to a social process of sharing.
- **Measurable positive outcome.** The point mechanism provides a measurable positive community aspect, and it is a way of quantifying the social capital existing in the community. Not only do points indicate that individuals have received answers to their questions (one explicit measurable outcome), but the social system as a whole can obtain a measurable positive outcome by indicating that the community as a whole is serving its own question-answering needs. Social capital in the community increases because the number of questions without answers remains low, and the satisfaction of questioners who obtain answers remains high. The more points that are exchanged in the system, the greater the indication that a community's needs are being met.

4 Examples of Socio-Technical Developments Promoting and Relying on Social Capital

Studying existing examples of social capital-sensitive systems in Section 3 in conjunction with the initial conceptual framework in Section 2 has provided the basis for our own system developments to *foster social creativity with social capital.* This section further illustrates this perspective by discussing how the understanding of social capital has influenced our system-building efforts and how, in return, our systems have enriched our understanding of social creativity and social capital.

4.1 *CodeBroker:* Evolutionary Construction of Reusable Software Component Repositories

CodeBroker [Ye 2001] is a system empowering software developers to share and reuse source code. Although the general belief is that software reuse improves both the quality and productivity of software development, systematic reuse has not yet met its expected success due to two basic challenging difficulties: (1) creating and maintaining a reuse repository, and (2) enabling software developers to build new software systems with components from the reuse repository and encouraging them to contribute to the repository. These two issues are in a deadlock: if software developers are unable to reuse, the investments on reuse cannot be justified; conversely, if companies are unwilling to invest in reuse, software developers have little to reuse.

We have taken a *social-technical* approach [Ehn 1988; Greenbaum and Kyng 1991; Grudin 1994] to address both issues simultaneously by fostering a reuse culture from the bottom up by encouraging software developers to reuse components from a repository that may not be of high quality in its initial state but can be evolved through the continual contributions of software developers during ongoing use of the repository. The concept of social capital helps us realize the important role of informal social networks in supporting reuse. To increase social capital to facilitate reuse and the evolutionary construction of reuse repositories, we are creating reuse repository systems that not only provide information on reusable components but enable developer communities to form, develop, and maintain a sense of shared identity around reuse repositories [Brown and Duguid 2000].

Our approach is guided by the SER model (see Section 2.3). In the *seeding* process, an initial reuse repository is created by clustering functionally related software components through the analysis of existing software systems, indexing the components, and providing an interface for users to access the components. This initial repository need not be of the highest quality but must provide opportunities for evolution. The initially seeded repository not only is the first incarnation of the intellectual capital owned by the software development organization, but also is an instantiated form of social capital of the organization because it creates the awareness for the expertise existing in the organization and a roadmap to the expertise distributed among software developers. Mediated by reusable components, knowledge sharing and indirect collaboration takes place, enabling software developers to create new systems by building upon each other's subsystems.

Software developers who attempt to reuse have to locate task-relevant components from an initially seeded reuse repository and to comprehend the located components. If no components that completely fit are located, or if the located components have some bugs to be fixed, software developers need to modify the components, and the modified components need to be contributed back to the repository through a sharing process, resulting in the *evolutionary growth* of the reuse repository.

A *reseeding* process is needed when the growth makes the repository too chaotic to grow further. During reseeding, the repository is reorganized, and its components are refactored and generalized, based on their use and the information added by software developers during the evolutionary growth.

Tools that support the easy location of reusable components are an essential factor to the success of our proposed approach to reuse due to two reasons: First, the goal of creating a reuse repository is to get components reused. Second, to involve software developers in the evolution of the reuse repository, they must first be made aware of the opportunities and benefit from reusing existing components in the repository because only when they enjoy certain benefits from the existing repositories will they reciprocate with their own contributions. Most current reuse repository systems [Mili, Mili et al. 1998] are created as systems independent of a software development environment. They are costly to use because software developers have to switch between development environments and reuse repository systems. Furthermore, because they support user-initialed information access mechanisms only, software developers who are not aware of the existence of reusable components end up reinventing the wheel. As the size of reuse repositories increases as a result of contributions made by software developers, it becomes increasingly difficult for software developers to anticipate the existence of such components.

Figure 2 shows a screen image of ***CodeBroker*** and its system architecture. ***CodeBroker*** consists of four software agents: Listener, Fetcher, Presenter, and Illustrator. Listener infers the needs for reusable components from the partially written program in the editing space (the top buffer) and creates a reuse query based on the inferred needs. The reuse query is passed to Fetcher, which searches the reuse repository to return a set of components that match the query. Presenter delivers the matching components to the developer in the delivery buffer (the low buffer). Presenter does not deliver all retrieved components; it uses discourse models to filter out task-irrelevant components and user models to filter out user-known components [Fischer and Ye 2001]. Discourse models that specify which part of the component repository is not of interest in the current development session, are created by developers during their previous interactions with the system to limit the search space. User models that contain those components that the developers know already are created by analyzing the programs that developers have written before.

CodeBroker helps developers utilize the existing social capital to seek help from other developers. When developers have difficulties in comprehending and applying the component of interest in their current task, they can invoke Illustrator, which searches a list of predetermined directories (such as those directories that contain programs written by peer developers) for a program that uses the component. Such an example provides context for understanding how to use the component [Fischer, Henninger et al. 1991]. Furthermore, the first line of the Example buffer reveals the author of the found example, which enables the creation of a knowledge transfer channel between the current developer who is experiencing difficulty and the developer who wrote the example. For such knowledge transfer to take place, the advanced developer must be willing to share his precious time and expertise, and the learning developer must have trust in the advanced developer. Both of the above issues rely on social capital, and the system creates a *connection* between the two developers that otherwise might not be associated. A successful collaboration between these two increases the sense of trust between them and makes later collaboration easier — important elements of social creativity and social capital.

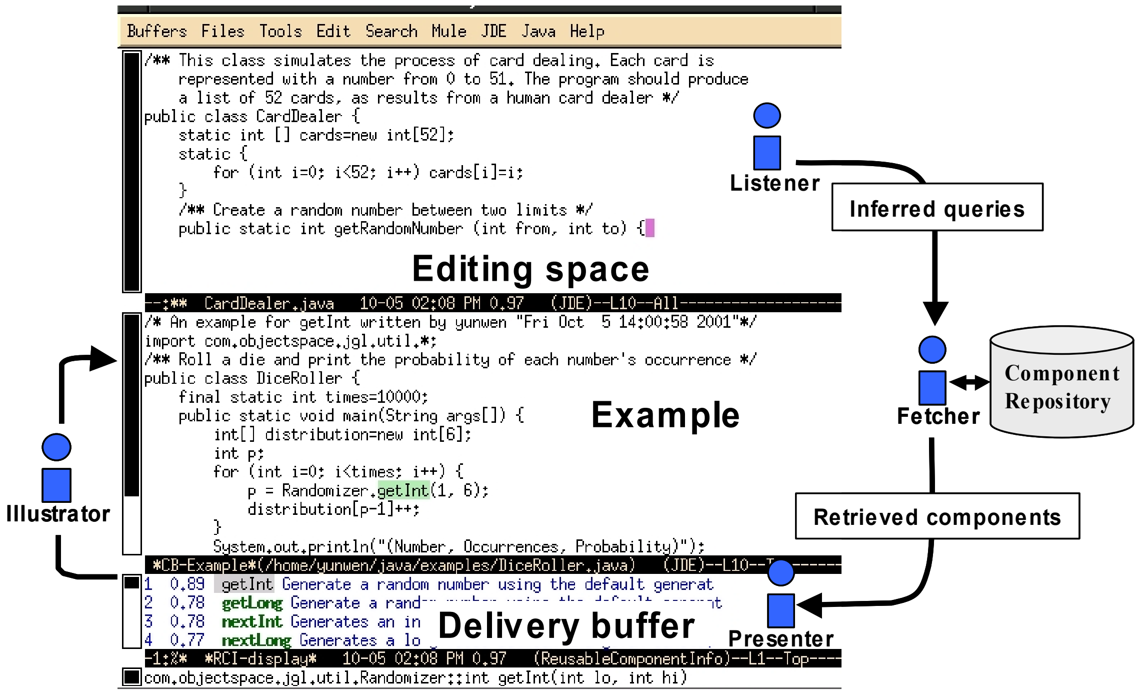

Figure 2: A Screen Image of *CodeBroker* and Its Architecture

CodeBroker increases the value of the reuse repository as a whole because it widens the reuse possibility of its components. At the same time, it increases the perceived value of components contributed by software developers through the sharing process because they know, with the help of ***CodeBroker***, their contribution will have a wider reach. Despite these motivating factors, a process of sharing requiring too much extra effort will inhibit software developers from doing so. We are current developing a new tool called *Enhancer* to reduce the cost of contribution by providing the following functionality:

- It automates as much as possible the process of packaging a component that an existing software developer decides to share with others for reuse through seamless integration with existing popular development environments. With the assistance of Enhancer, the software developer is able to contribute a component by invoking a high-level command such as "share it" within his or her normal development instead of carrying out detailed operations.
- It provides mechanisms to help software developers modify an existing component and then contribute the modified component back to the repository. The modification is made in the context of normal development so that software developers do not need to do too much extra work, and Enhancer automatically establishes a relationship between the newly modified component and the original component. Software developers who have contributed to the improvement of the component are acknowledged as a positive reinforcement to their reputations both as subject matter experts and as individuals who are willing to contribute.
- It captures the usage of reusable components by recording the developers who reuse a component to provide circumstantial evidence on the quality of the component, to create a list of subject matter experts that can potentially be tapped for help, and to explicitly recognize the contributions made by the original developer of the component and other contributors. Such data will give clear and measurable indications of how well knowledge sharing is taking place as reinforcement mechanisms for increasing social capital.

The ongoing enhancement of ***CodeBroker*** with tools such as *Enhancer* is driven by our conceptual framework and our assessment of earlier versions of ***CodeBroker*** [Ye 2001] to increase its potential to become a truly social capital-sensitive system.

4.2 The Envisionment and Discovery Collaboratory (EDC)

The ***EDC*** [Arias, Eden et al. 2000] is an environment in which participants collaboratively solve problems of mutual interest. The problem contexts explored in the ***EDC***, such as urban transportation planning, flood mitigation, and building design, are all examples of open-ended social problems. In these contexts, "optimal" solutions to problems do not exist, and the solutions depend on the participation of diverse stakeholders [Rittel 1984]. Solving problems in the ***EDC*** therefore requires development of social capital, and the technical and social features of the ***EDC*** are designed to enhance social capital. The ***EDC*** empowers users to act as designers in situated learning and collaborative problem solving activities. For most design problems, the knowledge to understand, frame, and solve them does not already exist, but is constructed and evolved during the solution process, exploiting the power of the "symmetry of ignorance" and "breakdowns." The ***EDC*** is an environment in which *social creativity* can come alive [Fischer 2000].

Figure 3 shows the ***EDC*** in use and illustrates some of these features. The ***EDC*** supports *face-to-face* problem-solving activities by bringing together individuals who share a common problem. The problem is discussed and explored by providing participants with a shared construction space in which participants interact with physical objects that are used to represent the situation currently being discussed. As users manipulate physical objects, a corresponding computational representation is updated by using technologies that recognize the placement and manipulation of physical objects. Computer-generated information is projected back on to the horizontal physical construction area, creating an augmented reality environment. This physical construction is coupled with information relevant to the problem currently being discussed.

The ***EDC*** is designed to *foster social creativity by increasing social capital* along the following dimensions:

Putting Owners of Problems in Charge. The collaborative design activities supported by the ***EDC*** rely upon the contribution and active participation of all involved stakeholders. Design domains consist of *ill-defined and wicked problems* [Schön 1983] in which there are no correct answers and in which framing problems is a major aspect of solving problems. Collaboration of stakeholders is an inherent aspect of these domains. For example, the citizens, urban planners, and transportation experts who solve or are influenced by urban transportation issues are themselves an integral part of the problem context. The goal of the ***EDC*** is to bring these stakeholders—each of them or each group owning one part of the problem—together to solve problems of mutual interest and to take active roles in addressing the problems that shape their lives.

Figure 3: The Envisionment and Discovery Collaboratory

Recognition and Awareness of Other Participants. The face-to-face nature of the ***EDC*** promotes recognition and awareness of other participants. By bringing people together around a table to solve a shared problem, the ***EDC*** attempts to recognize the importance of the various participants in the problem-solving context. Providing participants with a physical language to describe problems reduces the effort required to make the tacit knowledge of the participants explicit. This construction serves as a common reference point that allows participants to communicate *through* a shared representation, creating a neutral representation through which (potentially heated) discussion can take place [Arias 1988]. Computer recognition of physical representations is designed to allow for the computer to reduce the effort of capturing and formalizing problem information. Face-to-face discussions without some capture mechanism may be rich interactions, but only participants around the table benefit, and when the discussion is over the interaction is lost. Interaction with physical objects (and corresponding computational objects and models) provides a natural way to capture discussion by using formal structures that may be useful for indexing, history mechanisms, comparing multiple solutions, and helping to integrate diverse communication technologies.

Supporting Communities of Interest. Providing multiple avenues for participation is important because participants in the ***EDC*** may not share common backgrounds. They represent a *community of interest* [Fischer 2001], bringing together stakeholders from different domains who have different background knowledge and different things to contribute. The exchange of information is encouraged by providing stakeholders with tools to express their own opinions, requiring an open system that can accommodate and evolve based on new information. For example, city planners may bring formal information (such as the detailed planning data found in Geographic Information Systems), whereas citizens may use less formal techniques (such as sketching) to describe a situation from their points of view.

Impact of the SER Model. The ***EDC*** design is based on the SER model (see Section 2.3) to support this openness that is a necessity to let new aspects emerge. To reduce the effort of the stakeholders to contribute something, a particular ***EDC*** environment requires a substantial seed. Due to the emotionally and frequently politically charged nature of the problems being discussed, a seed requires significant sophistication before people are willing to utilize a tool in an authentic planning problem. Because many problems have significant time pressures, by the time a system can be adapted to the new situation, the problem may have already been resolved. Creating multiple avenues for participation requires close collaboration with the participants who will eventually need to use the system, and these collaborations are difficult to maintain if the act of helping create the seed is seen as additional work rather than as something that will prove beneficial immediately. Due to these challenges, assessing the ability of the ***EDC*** to enhance social capital in existing communities is a difficult task. To address these issues, we have created controlled scenarios with role play that lack many of the complexities of real-world problems but have the fundamental advantage that they can be duplicated and manipulated under various conditions [Eden, Hornecker et al. 2002]. We will use these participatory design experiments to gain a deeper understanding of how social creativity can be further enhanced by increasing social capital.

4.3 Courses-as-Seeds

Courses-as-seeds [dePaula, Fischer et al. 2001] is an educational model that explores social creativity and social capital in the context of fundamentally changing the nature of courses taught at educational institutions, primarily at the university level. Its goal is to create a culture of *informed participation* [Fischer and Ostwald 2002] that is situated in the context of university courses and yet extends beyond the temporal boundaries of semester-based classes. Traditionally, the content of a course is defined by the resources provided by instructors, such as lectures, readings, and assignments. By asking students to play an active role in assisting to construct their courses, courses rely not only on the intellectual capital provided by instructors but on the social capital provided by students. Courses are conceptualized as seeds, rather than as finished products, and students are viewed as informed participants who play an active role in defining the problems they investigate [Rogoff, Matsuov et al. 1998]. The output of each course contributes to an evolving information space that is collaboratively designed by all course participants, past and present.

An essential aspect of ***courses-as-seeds*** is the transformation of traditional classroom roles. Students act as active contributors—active not only in the assignments that are given to them, but also active in the design of the courses themselves. Instructors' roles are likewise transformed from a "sage on the stage" to a "coach on the side." The relationship network connecting students becomes important in fostering collaborative construction. We have developed and used *social capital-sensitive collaborative technologies* to provide opportunities to bring social creativity alive relying on our experience with open source projects (see Section 3.1 and [Scharff 2002]). Students chose their own projects and formed teams based on personal interests. These projects were linked together by the project leader (in our case, one of the instructors). Projects involved frequent concrete public deliverables, not in the form of progress updates and final reports that are detached from the actual work of the project, but to make the work on the project itself visible to team members. The students collaboratively constructed a course information environment using the *Swiki collaborative technology* [Rick, Guzdial et al. 2002].

Courses-as-seeds is grounded in the SER model (see Section 2.3). The *seed* for a course (embedded in the course information environment) is a set of themes, concepts, readings, and other materials, initially identified by instructors. Throughout the semester, students participating in the course (which includes face-to-face class meetings involving presentations and group discussion) *evolve* the course information environment. *Reseeding* involves the reorganization of the course information environment to become more useful for future generations of students, and to form the seed for future classes. In order to not overburden students with too much engagement (requiring *unrealistic levels of "social capital"*), we defined different roles on a rotating basis for specific activities. For example, some students answered homework assignments, whereas other students read these submissions, summarized them, reorganized them, and created summaries that represent a single distilled document for future readers (rather than requiring reading all previously posted assignments). These analyses formed a small kind of reseeding. In course projects, teams created public "home pages" (similar to the project home pages found in open source projects) for potential use by future students.

Transforming courses from traditional models of information transmission to collaborative construction requires a transformation of *mindsets* for students and instructors [Fischer 1999]. In an educational culture in which collaboration is stifled and often considered as cheating [Norman 2001], the standard expectations of students often hinder development of social capital. We encouraged group work through participation in the course information environments, creating group projects, and rewarded students for building on others' work. Nevertheless, many students resisted this collaborative construction, which creates a formidable challenge when students with "negative" social capital are asked to participate. We have found that self-selection is important for course success, discouraging students who did not wish to collaborate from participating. An important question remains whether it is more desirable to attempt to "force" collaboration (potentially creating social capital where none existed before) or to cultivate existing social capital.

Public deliverables have played a major role in establishing social capital in our courses. Encouraging students to send their contributions to each other can be a source of social creativity. Students reported that public deliverables are a good motivator to make higher-quality contributions because poor contributions make them look bad in the eyes of their peers. This is similar to findings in open source, in which developers claim that public deliverables encourage higher-quality components because anyone can observe (and evaluate) the contribution. Students who contributed frequently and with more substance were recognized as leaders by other students. In peer evaluations, students seem to react strongly to the activity of their team members. Because every change to the course information environment is recorded along with the user making the change, it is possible for students to see who is contributing what. Public deliverables encouraged students to share, which is rare in team projects, but is common with open-source-like course projects. When one team created a home page design that other teams liked, the other teams quickly adapted their sites to match the model. This increased the quality of their own sites and can be seen as a source of social creativity. The members of the original team were flattered by (and were glad that they were credited for) the adoption of their ideas by other teams.

5 Assessment

The analysis of existing systems (see Section 3) coupled with experiences derived from our own systems (see Section 4) has led to a greater understanding of the challenges involved in fostering social creativity with *social capital-sensitive systems* in collaborative design. Table 1 provides a summary statement of our findings.

Table 1: Summary of Social Creativity and Social Capital Aspects of the Environments Discussed

	Brief Description	Social Creativity	Social Capital
Open Source	approach and methodology for collaborative design and construction of software	source code availability breaks down barriers between users and designers; decentralized model encourages contributions by many	open process recognizes contributors and motivates them; public deliverables create objects for discussion and encourage high-quality deliverables
Experts Exchange	web-based knowledge sharing environment	high-quality answers encouraged by points and competition; website is collaborative evolved by both questioners and answerers	point system rewards participation; awareness of who does what; everyone wins: folks get answers, experts get recognition
CodeBroker	reuse environment	reuse of components leads to better software; environment encourages sharing and building on stable subcomponents	trust in developers leads to trust in code; willingness to help; reciprocity
Envisionment and Discovery Collaboratory (EDC)	face-to-face support for collaborative design	contribution of stakeholders having different perspectives and knowledge; solutions to problems emerge from discussions and individual design activities; support for the owners of problems	low barriers to contribute based on integrated and domain-oriented environment; owner of problems can directly engage in personally meaningful activities
Courses-as-Seeds	new methodology and computational support for learning and teaching	learning from each other is supported; instructors learn from students	students are motivated through personally meaningful projects; instructors must give up (some) control

5.1 Addressing the Adoption Barriers of Collaborative Technologies

Building a system that supports collaborative design is merely an enabling condition for fostering social creativity by increasing social capital. All too often, systems are created but the expected use never matches reality, which can be characterized by *"build it—and they will not come."* Analyses of failures of adoption of groupware applications [Grudin 1994] point to many complex factors, from institutional buy-in of proposed technologies, to involvement of system users during development. We will analyze these barriers using the equation

"utility = value / effort"

meaning that people will decide on the worthiness of doing something (utility) by relating the (perceived) value of an activity to the (perceived) effort of doing it.

Applying this equation, one technique for increasing utility is to *decrease the effort necessary to contribute to a system*. Capturing interactions automatically [Hill, Hollan et al. 1992; Adachi 1998] is a technique that decreases the effort needed to contribute. Giving participants an extensive *seed* rather than a completely empty open system helps provide immediate value and diminishes effort by helping ground new contributions by following the example set by the initial organization of the seed. *Incremental* *formalization* [Shipman and McCall 1994] can reduce effort by allowing unstructured data to be added quickly, and then allow the more time-consuming formalization of the data to be added over time. This allows users to strike a balance between the added utility of increased formalized (and machine-interpretable) structures (such as increased search capabilities) and the cost of adding the information necessary for searching. Techniques such as *Latent Semantic Analysis* [Landauer and Dumais 1997] can be added to incremental formalization by helping to create new associations without additional efforts of formalization. By measuring the similarity between different texts, latent semantic analysis can create associations between unstructured text documents without adding the human effort of adding structure or meta-data to that text. In the ***EDC*** (see Section 4.2), techniques such as physical interaction decrease participation effort by making the interaction more intuitive or natural than similar actions performed using traditional computer input methods of mice and keyboards. In ***CodeBroker*** (see Section 4.1), analyzing the formal structure of the current problem (such as code comments and program structure) provides usage data that greatly reduces the difficulty in formulating complex queries. In addition to these technical interventions, social aspects of communities can help to reduce the perceived effort of contributions. As contributions increase, familiarity with systems and with the community will also increase, and users will see evidence that contributing may not be too difficult. A sense of peer pressure will be created: If others can find time to contribute, then they can as well. For this reason, the *awareness* of the contributions of others and the positive acknowledgment of their accomplishments are critically important.

Increasing perceived value requires an understanding of what an individual and community will find valuable. Timely information that will help solve immediate problems will encourage people to associate contributions to this effort with something from which they can see an immediate benefit. Possible financial rewards from such information is one possibility [Andrews 2002], but not all valuable efforts will lead to them. In our ***courses-as-seeds*** model, students' participation (and any tangible outcome stored in online repositories) and the grades they receive are directly connected, so each contribution potentially adds value through the improvement of their grades. In open source, the realization that problems are intellectually interesting or fun helps to contribute to the perception of value in participating. Other community members are another important source of value: if someone can interact with a large number of people who can help to solve a problem (see the ***Expert Exchange*** in Section 3.2), then the community will create value. Similarly, if participation gives access to a group of people with whom someone would like to work, their presence will create value for that individual.

An interesting question raised by studying social capital is whether the notions of perceived utility and effort exist solely in the heads of individual participants or whether group activity creates an emergent and collective conceptualization of these ideas. As we have seen, social interactions play a role in individual perceptions. *Is there a similar way that groups as a whole have notions of effort and utility?* Systems such as the ***Experts Exchange*** indicate that this may be the case, as the participation of a large community of experts lowers the effort of any single individual and increases value due to the desire for a whole group of experts to collaborate.

5.2 Motivation

When an initially seeded system is adopted by its users, the further development and evolution of the system through the thriving of social creativity depends mainly on the *motivation* of users to contribute and participate, as well as the trust that users place in the quality of contributions made by other people. The relational dimension of the social capital definition provides a framework to assess the roles that motivation and trust play in fostering social creativity. Nahapiet and Ghoshal [Nahapiet and Ghoshal 1998] argue that the obligations of generalized reciprocity, shared norms and values of the community, mutual trust among members, and identification with the community are the four critical components to the relational dimension of social capital. Without active contribution and participation from motivated users, the process of collaborative knowledge construction will not succeed. Factors that affect users' motivation to contribute and participate are both cognitive (intrinsic) and socio-technical (extrinsic).

The precondition for motivating users to contribute to systems such as evolving reuse repositories or to participate in open source systems is that they must derive an intrinsic satisfaction in accomplishing their tasks and goals. In other words, the activity of contribution and participation itself must be perceived as engaging, satisfying, and pleasurable enough to be worth its associated efforts. The satisfaction may come from the intellectual challenge inherent in the task, the feeling of accomplishment and fulfillment, the joy of mastery of new knowledge, or the desire for social acceptance by identifying with some communities and associating with their members.

Social factors that positively reinforce intrinsic motivation include generalized reciprocity, social recognition, and rewards within a community.

Generalized reciprocity. In a community with a good stock of social capital, generalized reciprocity [Putnam 1995] is considered a social norm. This encourages members to do their own share of contributions in the expectation of receiving reciprocated help from other members. In our evolutionary reuse repository project, the ***CodeBroker*** system is designed to create an opportunity for users to obtain immediate benefits from the existing reuse repository through its delivery mechanisms. To abide by the social norm of generalized reciprocity, users are more likely to feel obliged to return the favor by reciprocating with their own work for the benefit of others [Nahapiet and Ghoshal 1998]. In ***open source*** communities, software developers receive assistance in testing, debugging, and functionality enhancing from other developers. In the ***EDC***, participating users learn new knowledge by receiving constructive critiques from other knowledgeable participants. Achieving social creativity requires the active participation of all stakeholders that have different perspectives and knowledge. The norm of generalized reciprocity motivates each stakeholder to contribute his or her own knowledge and understanding relevant to the common problem faced by all members, in the expectation of receiving reciprocated contributions from others who have different expertise and insights. As a result, the combined contributions can solve a problem that no individual member can solve alone. Recognizing the necessity for collaboration motivates some members to contribute at first. As members start to contribute, the sense of peer pressure causes other members to contribute reciprocally, gradually leading to the establishment of generalized reciprocity in the community.

Social recognition. Contributors' motivations increase when the community explicitly recognizes the effort and especially the value of user contributions. For example, listing the most active contributors in the ***Experts Exchange*** produces a quantified form of satisfaction and creates an informal competition for recognition as a subject matter expert. In open source communities, the existence of a large number of users provides psychological and social support for voluntary developers similar to the support an audience gives to theatrical performers [Nakakoji, Yamamoto et al. 2002]. Such explicit recognition mechanisms create community-wide awareness of the existence of experts and the distribution of expertise that facilitates potential information exchange and cooperation, therefore increasing the social capital of the community.

Rewards. Rewards for user contributions may come from both inside and outside the community. Active contributing users increase their own stock of social capital within the community by increasing their name recognition and their reputation as subject matter experts and willing helpers and collaborators (or good citizens). In return, they command respect and trust from other members and have the capability of executing larger influences in the community. Peer respect and trust make it easier for them to attract collaborators to advance their personal agenda in the future. We have found a strong correlation between user contributions and influential powers in open source communities. Active contributing developers gradually move toward the center of the community and play increasingly important roles in making decisions about further development [Nakakoji, Yamamoto et al. 2002]. If a user is recognized as an expert within an open source community, this reputation can spread to society at large, earning the user social-economical benefits, such as higher salary or social status. Some active open source developers have become internationally renowned for their strong technical skills and great insights, and some have benefited from the reputation they have built in open source communities to obtain highly paid jobs or provide consulting services to companies for profit. Because such external recognition depends on the influence the community as a whole could have on society, members have indirect and potential personal interests to behave as a good citizens to sustain the community.

All the above factors are components of social capital. Therefore, by taking social capital into consideration during the design of collaborative computer systems that facilitate social creativity, the utility of contributions increases due to the increase of perceived value, leading to higher motivation for active contribution and participation.

Technology can either promote or thwart a user's motivation to contribute and participate because it can either increase or decrease the effort of contribution. In terms of motivation to active contribution or participation, individual differences exist; some participants are more motivated than others. The technological barriers to contributing might thwart those less motivated. If the technology difficulty is extremely high, even the most motivated members might not be able to contribute. For example, it is impossible for extremely motivated and well-skilled users to contribute to the development of closed systems. If the technology requires no extra effort to contribute, no motivation is required to participate—users become active contributors indirectly through the work that they perform. In order to avoid making unreasonable demands on social capital, it is critical to design collaborative computer systems that are consistent with the *normal work process of users*, so that little extra work is required to contribute, allowing the population of motivated contributors to increase.

The decreased difficulty enables *more* people to participate, enhancing the social capital of the community by increasing the number of participants and enriching the connectedness of the members. The increased number of participating users means that each contribution can benefit more people through the enriched connectedness, and therefore, the value of contribution increases, leading to higher motivation for contributing.

5.3 Trust

As a key benefit of social capital, *mutual trust* among members of a community establishes better knowledge-sharing conduits, lowers transaction costs, resolves conflicts more easily, and creates greater coherence of collective actions. Because "*trust is the expectation that arrives within a community of regular, honest and cooperative behavior, based on commonly shared norms on the part of other members of that community* [Fukuyama 1995]," it develops when there is a history of favorable past interactions among those members [Preece 2002]. Trust leads community members to expect positive future interactions and encourages further interactions and participation.

In the context of collaborative computer systems that support social creativity, it is important that the initially seeded system have relatively coherent, stable, and predictable behaviors for users to develop an initial sense of trust in the system. By being able to manipulate familiar physical objects and receive immediate reactions from the system through simulation, users of the ***EDC*** transfer their trust in real-world objects into the system. Having a ready-to-run system in open source is the first step for its users to determine the trustworthiness of the system through firsthand experience by observing its behaviors. The ability to fully access and freely change source code reinforces the sense of trust by assuring the smooth resolution of future uncertainties because users have control over the system when it behaves unexpectedly or unsatisfactorily. Impersonal trust [McKnight, Cummings et al. 1998] in the system is transferred into interpersonal trust in the system developer or developers, which again is transferred to impersonal trust in the subsequent systems built by them. We are currently replicating this trust model in our attempt to create evolutionary reuse repositories through collaboration.

During the evolutionary growth phase, the initial trust develops or decays, depending on whether the continuing and repeated interactions between users and the system and between users match the expectation of the interacting partner, either the system or another user. In the ***EDC***, interpersonal trust among users develops naturally when they are engaged in face-to-face meetings, relying on their everyday life experiences. In virtual communities created in open source and reuse repositories, where face-to-face meetings rarely happen, the system needs to provide extra support to address the trust issue. Because users do not directly interact with each other, trust among them has to be mediated through common objects such as source code and reusable components that are in common use. The original developers of open source systems are generally trusted by other members due to the reasons discussed in the previous paragraph. Interpersonal trust in members of an open source community is transferred from the impersonal trust in the quality of their code contributions made over a relatively long period of participation in the community. The quality of code contributions is judged by execution and inspection of the code. However, this model does not work well in the case of reuse repositories for two reasons. First, most reusable components are not immediately executed and their quality cannot be easily judged. Second, careful inspection of the code requires a lot of time, which conflicts with one of the primary benefits of reuse: to speed up the development. Our approach to address the trust issue is to look at the *social context* of components [Brown and Duguid 2000]: who produces it, how many times it has been reused, who has reused it, and what people who have reused it say about it. This approach aims to develop a trust model that provides circumstantial evidence of the quality of a component based on the interpersonal trust among members in the community. Although the software developer may encounter the reusable component for the first time, the history of favorable past interactions between other trusted members and the component can help him or her trust it. The interpersonal trust and impersonal trust are mutual dependent. If the impersonal trust in a component based on interpersonal trust generates positive interactions, the interpersonal trust is positively reinforced; otherwise, the interpersonal trust decreases.

5.4 Evolution by Designers and Evolution by Users

To support social creativity, we need a new conceptual model to understand the relationship between information repositories that are the place to store information and knowledge and their designers and users. Figure 4 describes two general models of creating and using an information repository. Model (a) requires a thick, good input filter, which can be applied to select important and reliable information or to select a few dedicated information producers of high caliber, resulting in a relatively small information repository that contains only information of good quality but often misses potentially useful information. Model (a) supports information consumers, who are mostly passive, to locate and choose what they need from such an information repository. It is suited for static information repositories and is effective in domains in which changes are slow. It is not suitable, however, for evolutionary domains that require social creativity.

Model (b) describes the collaborative construction of information repositories. It has a thin input filter that allows not only dedicated producers but also active consumers (or local developers [Nardi 1993]) who are able and willing to contribute to put information into the information repository, resulting in a large information repository. This model requires a good, thick output filter that can provide information contextualized to the task at hand and the background knowledge of individual users [Ye 2001]. Information repositories created in Model (b) support social creativity based on the following reasons:

- they are information spaces owned by the people and communities who use them to do work, not by management or an IT department;
- they support the collaborative and evolutionary design of complex systems by allowing users to become active contributors;
- they are open and evolvable systems, serving not only as repositories of information, but also as mediums of communication and innovation;
- they are information spaces that can be evolved through many small contributions made by many people rather than large contributions by few people (as has been the case for knowledge-based systems of the past following Model (a)).

Purely self-organized (decentralized) evolution of complex artifacts and information repositories is a myth, however. For example, although open source software is constructed in a bazaar style—any user can change its source code—the main version for mass distribution is still controlled by a core set of "project leaders" who have the final say on the course of the evolution of a project. These people perform

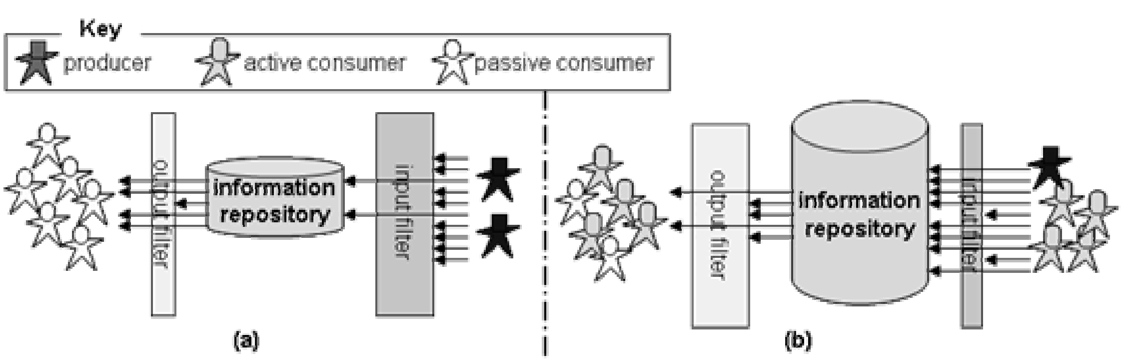

Figure 4: Two General Models of Creating and Using an Information Repository

centralized integration of information (during the reseeding phase) that is contributed by others in a decentralized manner (during the evolutionary growth phase). Contributors are explicitly acknowledged and often assume responsibility for the evolution of their subsystem. Open-source projects have many varieties of control structures, but each project has *some* centralized responsibility — none of them relies totally on decentralized evolution.

The evolution of living information repositories must have elements of both decentralized evolution and centralized integration. The mix of these modes, as well as the means of selection of individuals to assume roles of responsibility, takes many forms. The goal of making systems modifiable by users does not imply transferring the responsibility of good system design to the user. In general, modifications that normal users make will not be of the quality of those a system designer would make. Users are not concerned with the system per se, but in doing their work. On the other hand, users are concerned with the adequacy of the system as a tool to do *their* work, and as such they experience the fit, or misfit, between their needs and the capabilities of the tool. This is knowledge that the designer lacks because the designer does not use the tool to do work. Such systems do not decrease the responsibility or importance of the system designers, but shift the design emphasis on a finished system at design time to a system that can be adapted and modified at use time, by both users and designers.

Sustaining the usefulness and usability of evolving information repositories over time involves important challenges and trade-offs, as summarized in Table 2. Such factors depend upon whether these information repositories are evolved by specialists, who do not actually use the systems to do work, or evolved in the working context by knowledge workers, who are owners of problems and who evolve the environments in the context of their work.

Table 2: Information Repositories Evolved by Specialists versus Evolved in the Working Context

	evolved by designers	**evolved by users in the working context**
Examples	digital library of ACM	Reference Libraries and Websites of research groups, Eureka (Xerox Copier Technicians) [Brown and Duguid 2000]
Nature of individual entries	database-like entries	narratives, stories, best practices, cases
Economics	requires substantial additional resources (e.g., clerical support)	puts an additional burden on the domain workers
Cognitive dimensions	possible in domains in which objects are well-defined and tasks can be delegated	problem owners need to do it because the objects are emerging products of work
Motivation	work assignment	social capital

5.5 Self-Application

Building tools to support communities is not restricted to settings in which designers facilitate the work of other people. We have attempted to design our own community and our own tools to be *social capital-sensitive*. The advantage of this approach is greater than simply having an easily available community of users. *Self-application* allows us to see if ideas and tools work for us—if we are not willing to adopt them, then why should others? Another advantage is breaking down the barriers between designers of interactive experiences and "end-users" subjected to that experience. We wish to create a community in which all can help shape the experience, and the value gained by the individual to contribute is greater than the effort expended. Ideally, the community as a whole will strengthen the ties that bind it together and create social capital.

Although it is difficult to measure our group's social cohesion, emergent participation, and facilitation of participants directly, it is possible to understand some of these issues indirectly through our experience with collaborative technologies that we have both designed and used. The adoption challenges with organizational memories and collaborative work discussed above have been observed in our own work as well. Our key finding from this experience has been that *collaborative socio-technical environments require more than new technologies*. Collaborative technologies will not themselves change people's perceptions of technological barriers: people still must perceive a direct benefit for adopting new technologies, and they must gain more from contribution than what is required. The effort to contribute to a system must be minimal so that it will not interfere with getting the "real work" done. Social structures and work expectations have to change as well in a collaborative work environment. As one of our students observed, *"collaborative systems will not work in a non-collaborative society."* In our courses, if students do not want to contribute, or do not have a preconception that collaborative activity is expected or desirable, no amount of technology is going to change their lack of participation. Mindsets cannot be changed easily [Fischer 1999]. Work practices must change so that they are perceived as collaborative, involving a need for people to share authority and power, and breaking down traditional barriers (as well as personal barriers) [Zuboff 1988]. Not only do we rely on each other more than before (increasing the dependencies required to complete tasks), we do not always clearly communicate the expectations that are necessary from other people. Breaking down barriers of authority and control leaves individuals unclear about their precise roles, which are more obvious and more clearly defined in a closed system. As roles are transformed, so too must we transform both our understanding of our roles, and our ability to effectively negotiate who will do what.

6 Conclusions

Solving complex design problems requires the collaboration of numerous individuals with complementary skills. The resources necessary to solve problems are distributed among members of the community, and creative solutions emerge out of collaborative work. Collaborative systems need to be designed to facilitate social creativity, relying on the participation of members of the community and encouraging collaborative construction. The potential rewards of facilitating social creativity are great: encouraging participants to contribute to problems that shape their lives, distributing work so that individual contributions can be small but combined output can be substantial, leveraging the unique skills of individuals to create solutions that overlap mutual complementary competencies, and helping to create new ideas that would not have been created in isolation. Creating socially creative activities requires *social capital-sensitive environments*. All the resources of individuals and social groups, including the networks that connect people, the motivation and trust between members of the community, and the shared understanding and interests that bind the communities together, must be both utilized and cultivated.

Future generations of collaborative systems that wish to support social creativity can benefit from an increased understanding of the many aspects that affect social capital. A socio-technical perspective is important because new media, technology, approaches to collaboration, and communities must co-evolve in order to foster a richer form of collaboration. Just as technologies and documents provide structure, which influences contribution, so too must new technologies be designed to better support the kinds of contribution of a particular community. Designing for and increasing social capital involves a process of continuous evolution. The SER process model that encourages continuous change highlights the different challenges faced in creating initial seeds for collaboration, handling incremental changes, and periodically reorganizing and revising the contributions.

Increasing social capital requires an understanding not only of a group's internal cohesiveness, but what role individuals play in the formation of that group. Individual participation is a necessary precondition for group work, and that participation relies on the balance between the perceived benefit of a contribution to the perceived effort of becoming involved. Personal motivations, individual relationships, expectations from participation, recognition by peers, and tangible metrics for success are important for participants. Technology can be used not only to decrease the potential effort made by individuals, but can bring together people, and by bringing them together can increase the perceived value of participation. The collective activity and sharing culture of a community can be amplified by technologies that encourage individuals to participate in a group effort. An important challenge for the next generation of collaborative systems is to acknowledge the individual and collective forces required in collaborative design and to foster social creativity by increasing social capital.

7 Acknowledgments

The authors thank the members of the Center for LifeLong Learning & Design at the University of Colorado, who have made major contributions to the conceptual framework described in this paper. The research was supported by (1) the *National Science Foundation*, Grants (a) REC-0106976 “Social Creativity and Meta-Design in Lifelong Learning Communities”, and (b) CCR-0204277 “A Social-Technical Approach to the Evolutionary Construction of Reusable Software Component Repositories”; (2) *SRA Key Technology Laboratory*, Inc., Tokyo, Japan; and (3) the *Coleman Initiative*, San Jose, CA.

8 References

Adachi, T. (1998). *Utilization of Usage Data to Improve Organizational Memory*, M.S. Thesis, Department of Computer Science, University of Colorado at Boulder, Boulder, CO.

Andrews, D. C. (2002). "Audience-Specific Online Community Design." *Communications of the ACM* **45**(4): 64-68.

Arias, E. and G. Fischer (2000). *Boundary Objects: Their Role in Articulating the Task at Hand and Making Information Relevant to It*. International ICSC Symposium on Interactive & Collaborative Computing (ICC’2000), University of Wollongong, Australia, ICSC Academic Press, Wetaskiwin, Canada: 567-574.

Arias, E. G. (1988). "Bottom-up Neighborhood Revitalization: A Language Approach for Participatory Decision Support." *Urban Studies* **33**(10): 1831-1848.

Arias, E. G., H. Eden, G. Fischer, A. Gorman and E. Scharff (2000). "Transcending the Individual Human Mind—Creating Shared Understanding through Collaborative Design." *ACM Transactions on Computer Human-Interaction* **7**(1): 84-113.

Bourdieu, P. (1983). Forms of Capital. *Handbook for Theory and Research for the Sociology of Education*. J. G. Richardson. New York, Greenwood Press**:** 241-258.

Brown, J. S. and P. Duguid (2000). *The Social Life of Information*. Boston, MA, Harvard Business School Press.

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

Campbell, D. T. (1969). Ethnocentrism of Disciplines and the Fish-Scale Model of Omniscience. *Interdisciplinary Relationships in the Social Sciences*. M. Sherif and C. W. Sherif. Chicago, Aldine Publishing Company**:** 328-348.

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

Claypool, M., D. Finkel and C. Wills (2001). *An Open Source Laboratory for Operating Systems Projects*. 6th Annual Conference on Innovation and Technology in Computer Science Education, Canterbury, United Kingdom: 145-148.

Cohen, D. and L. Prusak (2001). *In Good Company: How Social Capital Makes Organizations Work*. Boston, MA, Harvard Business School Press.

Coleman, J. C. (1988). "Social Capital in the Creation of Human Capital." *American Journal of Sociology* **94**: S95-S120.

Davis, M., W. O'Donovan, J. Fritz and C. Childress (2000). *Linux and Open Source in the Academic Enterprise*. Conference on User Services: Building the Future, Richmond, VA: 65-69.

dePaula, R., G. Fischer and J. Ostwald (2001). *Courses as Seeds: Expectations and Realities*. Proceedings of the Second European Conference on Computer-Supported Collaborative Learning (Euro-CSCL' 2001), Maastricht, Netherlands: 494-501.

DiBona, C., S. Ockman and M. Stone, Eds. (1999). *Open Sources: Voices from the Open Source Revolution*. Sebastopol, CA, O'Reilly & Associates.

Eden, H., E. Hornecker and E. Scharff (2002). *Multilevel Design and Role Play: Experiences in Assessing Support for Communities of Interest*. Proceedings on Designing Interactive Systems (DIS '02): Processes, Practices, Methods, and Techniques, London, England: 387-392.

Ehn, P. (1988). *Work-Oriented Design of Computer Artifacts*. Stockholm, Arbetslivscentrum.

Engeström, Y. (2001). "Expansive Learning at Work: Toward an Activity Theoretical Reconceptualization." *Journal of Education and Work* **14**(1): 133-156.

Erickson, T. and W. A. Kellogg (2001). Social Translucence: Designing Systems That Support Social Processes. *Human-Computer Interaction in the New Millennium*. J. M. Carroll. New York, ACM Press**:** 325-345.

Experts-Exchange (2002). The Experts Exchange. **2002.** <http://www.experts-exchange-com>

Fischer, G. (1998). *Beyond 'Couch Potatoes': From Consumers to Designers*. Asian Pacific Computer Human Interaction Conference (APCHI'98), Tokyo, IEEE Computer Society: 2-9.

Fischer, G. (1999). Lifelong Learning: Changing Mindsets. *7th International Conference on Computers in Education on "New Human Abilities for the Networked Society" (ICCE'99, Chiba, Japan)*. G. Cumming, T. Okamoto and L. Gomez. Omaha, IOS Press**:** 21-30.

Fischer, G. (2000). "Social Creativity, Symmetry of Ignorance and Meta-Design." *Knowledge-Based Systems Journal (Special Issue on Creativity & Cognition), Elsevier Science B.V., Oxford, UK* **13**(7-8): 527-537.

Fischer, G. (2001). *Communities of Interest: Learning through the Interaction of Multiple Knowledge Systems*. 24th Annual Information Systems Research Seminar In Scandinavia (IRIS'24), Ulvik, Norway, Department of Information Science, Bergen, Norway: 1-14.

Fischer, G. and A. Girgensohn (1990). End-User Modifiability in Design Environments. *Human Factors in Computing Systems, (CHI'90) (Seattle, WA)*. New York, ACM**:** 183-191.

Fischer, G., J. Grudin, R. McCall, J. Ostwald, D. Redmiles, B. Reeves and F. Shipman (2001). Seeding, Evolutionary Growth and Reseeding: The Incremental Development of Collaborative Design Environments. *Coordination Theory and Collaboration Technology*. G. M. Olson, T. W. Malone and J. B. Smith. Mahwah, NJ, Lawrence Erlbaum Associates**:** 447-472.

Fischer, G., S. R. Henninger and D. F. Redmiles (1991). Cognitive Tools for Locating and Comprehending Software Objects for Reuse. *Thirteenth International Conference on Software Engineering (Austin, TX)*. Los Alamitos, CA, IEEE Computer Society Press**:** 318-328.

Fischer, G. and J. Ostwald (2002). *Seeding, Evolutionary Growth, and Reseeding: Enriching Participatory Design with Informed Participation*. Proceedings of the Participatory Design Conference (PDC '02), Malmö University, Sweden.

Fischer, G. and E. Scharff (2000). *Meta-Design—Design for Designers*. 3rd International Conference on Designing Interactive Systems (DIS 2000), New York, ACM: 396-405.

Fischer, G. and Y. Ye (2001). Personalizing Delivered Information in a Software Reuse Environment. *Proceedings of User Modeling 2001 (8th International Conference, UM 2001), Sonthofen, Germany*. M. Bauer, P. Gmytrasiewicz and J. Vassileva. Heidelberg, Springer Verlag**:** 178-187.

Florida, R. (2002). *The Rise of the Creative Class: And How It's Transforming Work, Leisure, Community, and Everyday Life*. New York, NY, Basic Books.

Fogel, K. F. (2000). *Open Source Development with CVS*. Scottsdale, AZ, The Coriolis Group.

Fukuyama, F. (1995). *Trust: The Social Virtues and the Creation of Prosperity*. New York, Free Press.

Fukuyama, F. (1999). *Social Capital and Civil Society*. IMF Conference on Second Generation Reforms, Washington, DC.

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

Grudin, J. (1994). "Groupware and Social Dynamics: Eight Challenges for Developers." *Communications of the ACM* **37**(1): 92-105.

Hanifan, L. J. (1916). "The Rural School Community Center." *Annals of the American Academy of Political and Social Science* **67**: 130-138.

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

Hill, W. C., J. D. Hollan, D. Wroblewski and T. McCandless (1992). Edit Wear and Read Wear. *Proceedings of CHI'92 Conference on Human Factors in Computing Systems*. P. Bauersfeld, J. Bennett and G. Lynch. New York, ACM**:** 3-9.

John-Steiner, V. (2000). *Creative Collaboration*. Oxford, Oxford University Press.

Landauer, T. K. and S. T. Dumais (1997). "A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge." *Psychological Review* **104**(2): 211-240.

Mackay, W. E. (1990). *Users and Customizable Software: A Co-Adaptive Phenomenon*, Sloan School of Management Dissertation, Massachusetts Institute of Technology, Cambridge, MA.

MacLean, A., K. Carter, L. Lovstrand and T. Moran (1990). User-Tailorable Systems: Pressing the Issues with Buttons. *Proceedings of CHI'90 Conference on Human Factors in Computing Systems*. J. Carrasco and J. Whiteside. New York, ACM**:** 175-182.

McKnight, D. H., L. L. Cummings and N. L. Chervany (1998). "Initial Trust Formation in New Organizational Relationships." *Academy of Management Review* **23**: 383-399.

Meyerson, E. (1994). "Human Capital, Social Capital and Compensation: The Relative Contribution of Social Contacts to Manager's Income." *Acta Sociologica* **37**: 383-399.

Mili, A., R. Mili and R. T. Mittermeir (1998). A Survey of Software Reuse Libraries. *Systematic Software Reuse*. W. Frakes. Bussum, The Netherlands, Baltzer Science**:** 317-347.

Mockus, A., R. T. Fielding and J. Herbsleb (2000). *A Case Study of Open Source Software Development: The Apache Server*. 22nd International Conference on Software Engineering, Limerick, Ireland: 263-272.

Nahapiet, J. and S. Ghoshal (1998). "Social Capital, Intellectual Capital, and the Organizational Advantage." *Academy of Management Review* **23**: 242-266.

Nakakoji, K., Y. Yamamoto, Y. Nishinaka, K. Kishida and Y. Ye (2002). *Evolution Patterns of Open-Source Software Systems and Communities*. Proceedings of International Workshop on Principles of Software Evolution (IWPSE 2002), Orlando, FL: 76-85.

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

Norman, D. (2001). In Defense of Cheating**.** <http://jnd.org/dn.mss/InDefenseOfCheating.html>

Oppermann, R., Ed. (1994). *Adaptive User Support*. Hillsdale, New Jersey, Lawrence Erlbaum.

Perens, B. (1999). The Open Source Definition. *Open Sources: Voices from the Open Source Revolution*. C. DiBona, S. Ockman and M. Stone. Sebastopol, CA, O'Reilly & Associates**:** 171-188.

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

Preece, J. (2002). "Supporting Community and Building Social Capital: Introduction." *Communications of the ACM* **45**(4): 37-39.

Putnam, R. D. (1995). "Bowling Alone: America's Declining Social Capital." *Journal of Democracy* **6**(1): 65-78.

Raymond, E. S. (1998). "Homesteading the Noosphere." *FirstMonday* **3**(10): <http://www.firstmonday.dk/issues/issue3_10/raymond/index.html>.

Raymond, E. S. (1999). *The Cathedral & The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA, O'Reilly & Associates.

Raymond, E. S. and B. Young (2001). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA, O'Reilly & Associates.

Reisberg, D. (1997). *Cognition*. New York, Norton & Company.

Rick, J., M. Guzdial, K. Carroll, L. Holloway-Attaway and B. Walker (2002). *Collaborative Learning at Low Cost: CoWeb Use in English Composition*. Conference on Computer Supported Collaborative Learning (CSCL 2002), Boulder, CO: 435-442.

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

Rogoff, B., E. Matsuov and C. White (1998). Models of Teaching and Learning: Participation in a Community of Learners. *The Handbook of Education and Human Development—New Models of Learning, Teaching and Schooling*. D. R. Olsen and N. Torrance. Oxford, Blackwell**:** 388-414.

Salomon, G., Ed. (1993). *Distributed Cognitions: Psychological and Educational Considerations*. Cambridge, United Kingdom, Cambridge University Press.

Scharff, E. D. (2002). *Applying Open Source Principles to Collaborative Learning Environments*. Conference on Computer Supported Collaborative Learning (CSCL 2002), Boulder, CO: 499-500.

Scharff, E. D. (2002). *Open Source: A Conceptual Framework for Collaborative Artifact and Knowledge Construction*, Ph.D. Dissertation, Department of Computer Science, University of Colorado, Boulder, CO.

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

Schuler, D. and A. Namioka, Eds. (1993). *Participatory Design: Principles and Practices*. Hillsdale, NJ, Lawrence Erlbaum Associates.

Shipman, F. and R. McCall (1994). Supporting Knowledge-Base Evolution with Incremental Formalization. *Human Factors in Computing Systems, INTERCHI'94 Conference Proceedings*. New York, ACM**:** 285-291.

Ye, Y. (2001). *Supporting Component-Based Software Development with Active Component Repository Systems*, Ph.D. Dissertation, Department of Computer Science, University of Colorado at Boulder, Boulder, Colorado.

Zuboff, S. (1988). *In the Age of the Smart Machine*. New York, Basic Books, Inc.