Socio-Technical Systems:
A Meta-Design Perspective

Gerhard Fischer, University of Colorado, Boulder, USA

Thomas Herrmann, University of Bochum, Germany

ABSTRACT

Meta-design of socio-technical systems complies with the need to integrate two types of structures and processes: technical systems, which are engineered to provide anticipatable and reliable interactions between users and systems, and social systems, which are contingent in their interactions and a subject of evolution. Meta-design is focused on objectives, techniques, and processes to allow users to act as designers. It provides, rather than fixed solutions, frameworks within which all stakeholders can contribute to the development of technical functionality and the evolution of the social side, such as organizational change, knowledge construction, and collaborative learning. This paper combines the theoretical framework of meta-design and its underlying principles with the consideration of methodological aspects and practical cases. Five different principles are explored: (1) cultures of participation, (2) empowerment for adaptation and evolution, (3) seeding and evolutionary growth, (4) underdesign of models of socio-technical processes, and (5) structuring of communication. Design collaboratories and knowledge management are used as examples to analyze meta-designed systems representing socio-technical solutions as well as frameworks within which socio-technical solutions can be developed. The combination of theoretical and methodological considerations leads to a set of practical guidelines for meta-designers.

Keywords: Collaboration, Cultures of Participation, Evolutionary Growth, Knowledge Management, Meta-design, Participatory Design, Reseeding Model, Semi-Structured Modeling, Socio-Technical Systems (STS)

INTRODUCTION

New technologies and new media are important driving forces and prerequisites to address the complex and systemic problems our societies face today. But technology alone does not improve social structures and human behavior, making the design of *socio-technical systems (STSs)* a necessity rather than an academic luxury.

A unique challenge faced in focusing on STSs is that that they combine two types of fundamentally different systems:

- *Technical systems* that are produced and continuously adapted to provide a reliable, anticipatable relationship between user input and the system's output. This relationship is engineered to serve the needs of users and is—at least incrementally—preplanned.

• *Social systems* that are the result of continuous evolution including emergent changes and behavior. The development of their characteristics cannot be planned and controlled with respect to the final outcome; the changes within STSs are a matter of contingency (Luhmann, 1995) and can only—if ever—be understood afterward and not in advance; social systems mainly serve their own needs and not those of others.

The strength of STSs is that they integrate these different phenomena so that they increase their performance reciprocally. Even more important, the integration of technical and social systems helps them to develop and to constitute each other, for example, the interaction among community members is supported by technical infrastructure, and the members themselves can contribute to the development of the infrastructure, as is typically demonstrated by open source communities. However, the relationships between the development of the social and the technical are not deterministic but contingent. For example, developing software for specific organizations does not deterministically change them but only influences the evolution of their social structures. Software designers can be reflective with respect to the impact of a software system on its social context, and they can make their assumptions about the expected evolution of the social system explicit and a matter of discourse, but they cannot control the organizational change.

One emerging unique opportunity to make a systematic and reflected contribution to the evolution of social structures in STSs is *meta-design* (Fischer & Giaccardi, 2006), representing a design perspective supporting the evolution of systems that have contingent characteristics. Whereas many design activities aim to develop concrete technical solutions, meta-design provides a *framework* within which STSs can be developed. Fischer and others (Fischer & Giaccardi, 2006) have outlined a variety of important characteristics

of meta-design. The most important principles characterizing a meta-design framework for the development of STSs are (Fischer, 2010):

1. Support for *cultures of participation* that put the owners of problems in charge and give them control of how technical systems are used and which functionality is underlying the usage. In this context, an ecology of roles (Preece & Shneiderman, 2009) will develop including developers, co-developers, consultants, facilitators, and curators (see the section, "Cultures of Participation").

2. Mechanisms to support *empowerment for adaptation* and *evolution at use time* by offering functionality for *tailorability*, *customization*, and user-driven adaptability (Mørch, 1997) (see the subsection "Empowerment for Adaptation and Evolution").

3. A procedure model that includes the phases of *seeding*, *evolutionary growth*, and *reseeding* (Fischer & Ostwald, 2002), in which the seed represents a result of underdesign—it represents basic structures and is in accordance with the relevant standards but it leaves space and options for the development of concrete details (see the subsection "Seeding, Evolutionary Growth, and Reseeding Model").

Herrmann et al. (2000, 2004) have conducted several *empirical studies* in which they have analyzed the relevance of communicational practices in the course of developing STSs. Herrmann (2009) describes a list of practical cases that support the methodological consideration in this paper. Based on an action research approach, Avison et al. (1999) have gradually developed methodological concepts that comply with the principles of socio-technical meta-design:

4. *Semi-structured modeling* to support and accompany the communication during the evolution of a socio-technical system. The models document requirements, plans, technical specifications, business

STSs, and processes on the one hand, and the specification of details on the other hand (see the subsection "Underdesign of Models of Socio-Technical Processes"). Semi-structured modeling is closely related to underdesign, which is an important principle of meta-design (Fischer, 2003).

5. *Walkthrough-oriented facilitation* as an example for the structuring of communication. It supports the integration of various perspectives, the negotiation of design decisions, the building of commitments about how technology will be used and adapted, and the evaluation of prototypes (see the subsection "Structuring of Communications").

The goal of this paper is to integrate these five conceptual principles under the perspective of meta-design of STSs. Focusing meta-design on the development and evolution of STSs gives the opportunity for a more detailed reflection of methodological implications and guidelines. Meta-design of STSs leads to new considerations that go beyond traditional participatory design, end-user-programming, or previous principles for the design of STSs (Cherns, 1976; Eason, 1988).

In our analysis, we draw on a body of literature that contributes to the clarification of socio-technical phenomena (Checkland, 1981; Mumford, 1987, 2000; Trist, 1981; Whitworth, 2009). Our analysis is based on a variety of concepts that stem from an interdisciplinary background, such as the interdependence between technology and organization (Orlikowski, 1992); sociological systems theory (Luhmann, 1995); wicked problems (Rittel & Webber, 1973); scenario-based design (Carroll, 1995); contingency (Pedersen, 2000); and participatory design (Kensing & Blomberg, 1998). This paper does not describe a complete set of tools and methods for the meta-design of STSs but rather describes the background of a meta-design methodology as well as examples of methods.

The theoretical background of STSs and meta-design are described in the next section. The third section gives a detailed description of the five principles of meta-design as they are listed above. These theoretical considerations are complemented with insights, as they can be derived from concrete empirical examples. The fourth section elucidates that there is a wide spectrum of software for which meta-design can be applied, and it continues by focusing on two typical areas of socio-technical meta-design, collaboratories and knowledge management (KM).

- *Collaboratories*, which have a clear location, include various competences and perspectives and various roles with respect to the development of technology, commitments, and organizational structures.
- *Knowledge management* within companies and communities includes various possibilities to build knowledge, to integrate it, to develop social relationships, and to identify appropriate technical support etc.

Based on the theoretical analysis and the reflection of practical cases, the fifth section provides a list of guidelines for the practice of meta-design. The concluding section summarizes the reasons for a meta-design approach in the context of socio-technical systems.

SOCIO-TECHNICAL SYSTEMS

Characteristics of STSs

Socio-technical systems can be understood as the systematic integration of two kinds of phenomena that have very diverging, partially contradictive characteristics. STSs are composed *both* of computers, networks, and software, *and* of people, procedures, policies, laws, and many other aspects. STSs therefore require the *co-design* of social and technical systems.

Whereas *technical systems* are purposeful artifacts that can reliably and repeatedly be used to support human needs and to enhance human capabilities, *social systems* are dedicated to purposes that lay within themselves and are a matter of continuous change and evolution,

which makes their behavior difficult to antici- pate. Social structures can be identified on sev- eral levels: communicative interaction between people or in small groups such as families or teams, organizations or organizational units, communities, or social networks. The reac- tions of social systems to their environment are contingent—they are not independent from external stimuli, but they also are not determined by them. As opposed to necessity, universality, constancy, and certainty, *contingency* (Pedersen, 2000, p. 413).

- Refers to variability, particularity, mutabil- ity, and uncertainty;
- Implies that the system creates its own necessity in its pattern of reactions toward events (Kirkeby, 2000, p. 11); and
- Provides a basis for continuous evolu- tion, including opportunities for emergent changes.

How new phenomena will emerge in social systems cannot be predicted or made the result of a well-planned, algorithmically organized procedure; they depend on coincidences and are context related in the sense of situatedness (Suchman, 1987). Technical systems may also react contingently toward their users, but the more mature a technical system has become, the more one will expect that it is reliable for the us- ers, predictable, and noncontingent. Obviously, the socio-technical perspective covers more aspects than the viewpoint of human-computer interaction (HCI): it is about the relationship between technical infrastructure as a whole and structures of social interaction, which cover organizational and coordination issues, sense making and common ground as a basis for communication, power relations, negotiation, building of conventions, and so forth.

It is not unlikely that formal communica- tion, anticipatable procedures, scripts, and prescriptions may be empirically observable within in social systems. For example, workflow management systems (Herrmann & Hoffmann, 2005) demonstrate the managerial attempt to implement scripts and institutionalize plan- oriented behaviour in the context of organiza- tions. However, it is a social system's dominant characteristic that rules and routines can be revised and become subjects of negotiation, and it cannot be predicted whether and when anticipatable behavior is no longer sustained but becomes a subject of evolutionary or emergent change.

By contrast to those researchers who as- sume that complex human activities can also be assigned to technical systems (Latour, 1999), we suggest that the crucial characteristics of social versus technical systems point in two opposite directions (Table 1). The basic differ- ences outlined in the table also apply to artificial intelligence applications and large networks of autonomous agents. The strength of socio- technical systems results of the integration of these two kinds of different phenomena.

Beyond Coincidental Connectedness: The Need for Systematic Integration

STSs are more than a coincidental connected- ness of technical components and people. ".. STS research is not just applying sociological principles to technical effects (Coiera, 2007), but [it explores, G.F., T.H.] how social and technical aspects integrate into a higher-level system with emergent properties" (Whitworth, 2009, p. 4).

The synergy between technical and social systems can be achieved only if both parts are closely integrated. One of the important theoretical challenges with respect to STSs is to explain how this integration can happen, by which factors it is influenced, and how it can be observed. Sociologists such as Luhmann (1995) and Habermas (1981) identify *com- munication*, amongst all kind of human activi- ties, as the most relevant constituent of social systems. Our research emphasizes the role of communication when we try to understand the integration between social and technical structures. The degree of integration between

Table 1. Main characteristics of technical and social systems

	Technical systems	Social systems
Origins	Are a product of human activity; can be designed from outside.	Are the result of evolution, cannot be designed but only *influenced* from outside.
Control	Are designed to be controllable with respect to prespecified performance parameters.	Always have the potential to challenge control.
Situatedness	Low: preprogrammed learning and interaction with the environment.	High: includes the potential of improvisation and nonanticipatable adaptation of behavior patterns.
Changes	Are either preprogrammed (so that they can be simulated by another technical system) or a result of intervention from outside (so that a new version is established).	Evolutionary: gradual accumulation of small, incremental changes, which can lead to emergent changes (which, however are not anticipatable). There is no social system that can simulate the changes of another social system.
Contingency	Are designed to avoid contingency; the more mature a version is, the less its reactions appear as contingent.	The potential for change and evolution is based on contingency.
Criteria	Correctness, reliability, unexpected, unsolicited events are interpreted as malfunction.	Personal interest, motivation; in the case of unsolicited events, intentional malpractice may be the case.
Modeling	Can be modeled by describing how input is processed and leads to a certain output.	Models can only approximate the real behavior and have continuously to be adapted.
Modus of development	Is produced or programmed from outside.	Develops by evolution that is triggered by communicative interaction.

social and technical structures increases with the extent of the following factors.

- Communication that uses the *technical systems as a medium* helps to convey communicational acts and shapes them.
- Communication *about the technical system* includes how it is used, how it has to be maintained, how it could be adapted to the needs of an organization and its users, how its effects can be compared with other technical systems, and so forth. This kind of communication leads to what we can call the appropriation of the technical system (Pipek, 2005) by the social system. The communication mirrors the organization's understanding of the technical structures.
- Content or social structures (e.g., responsibilities or access rights) *regulating communication* are being represented within the technical system as well as the social structures.

- Self-description describes and constitutes the characteristics of the STSs and can be found in the oral communication and in the documents of the social system as well as in the technical system's content and structures (Kunau, 2006).

With respect to the integration between technology and social structures, it is important to understand that technology is not mainly represented by artifacts such as hardware but by methods and procedures that are connected with these artifacts. These procedures and methods build the bridge between technology and communications in social interactions. The invention of writing is a typical example: the method of how to write is the dominating aspect compared with the means that help to make the written durable. Thus, the social impacts—such as shift of power and control, distributed cognition, shift in tasks, and so forth—are caused

much more by the methodological aspects of writing than by its physical materiality.

The need for seamless socio-technical integration is emphasized by many authors and approaches—for example, by Eason's (1988) or Cherns' (1976) principles of socio-technical design, by Kensing et al.'s (1996) MUST-Method, or Wulf and Rohde's (1995) approach of integrated organization and technology development.

The relevance of socio-technical integration can be observed in many areas, for example, knowledge management or computer-supported collaborative learning (CSCL); it is definitely insufficient just to introduce a document management system or to provide all schools with Internet access. Introducing a technical system is a necessary but not sufficient measure to be taken. They have to be complemented with interventions that aim on organizational as well as mental changes to promote the appropriation (Pipek, 2005) of the technology. Employees will not be willing to share their knowledge with others without role models and facilitation support, students will not learn more or be more motivated, and teachers will not teach better, as long as CSCL systems are not accompanied by new forms of educational experience.

Within the large set of areas where socio-technical integration takes place, this paper focuses on the design of technical systems that are related to information processing and software development. To determine a clear focus with respect to the social structures into which technical systems are integrated proves difficult. The classical socio-technical literature (Trist, 1981) usually addresses the meso-level, concerning such organizations as companies, administrations, and nongovernment organizations (NGOs) or their subunits. However, with the emergence of the web, and in particular Web2.0 and social software, phenomena have to be taken into account such as virtual communities, which form larger units between the meso- and the macro-level where individuals and/or several companies are interacting within new social structures that became possible only by new types of technical infrastructure. The

new phenomena that emerged in the context of the web and Web2.0 also gave new reasons for intensifying socio-technical analyses and approaches. It also became obvious that socio-technical phenomena cannot always be appropriately described by the concept of "system" as it is defined by older (von Bertalanffy, 1973) or newer (Maturana & Varela, 1980) systems theory. By contrast, it can be more adequate to focus the analysis on *socio-technical environments* (Carmien et al., 2005) within which the integration of technical and social structures can develop. Such a socio-technical environment is less the result of engineering or design activities and more a framework within which design takes place and is intertwined with the evolutionary growth of social structures (see the *intermediate level* of Table 2).

With respect to their evolution, socio-technical systems integrate two characteristics: on the one hand, they are the result of such human activities as design, engineering, managing, and communication; on the other hand, they serve on a higher level as the environment or framework within which these kind of human activities take place. Therefore we argue that the concept of "meta-design" is more appropriate to describe how socio-technical systems or environments are developed and do develop.

A CONCEPTUAL FRAMEWORK FOR META-DESIGN

Meta-design (Fischer & Giaccardi, 2006) is an emerging conceptual framework aimed at defining and creating socio-technical systems or environments and at understanding both as living entities. It extends existing design methodologies focused on the development of a system at design time by allowing users to become co-designers at use time. Meta-design is grounded in the basic assumption that future uses and problems cannot be completely anticipated at design time, when a system is developed (Suchman, 1987; Winograd & Flores, 1986). At use time, users will discover mismatches between their needs and the support that an

Table 2. A three-level model of meta-design

	Abstract description	**Examples**
Meta level Beliefs and concepts of meta-design	Meta-design provides a philosophy—a set of beliefs and guidelines—that helps to select appropriate methods and procedures. It is substantiated by theoretical insights and by concrete empirical examples.	Orientation on a culture of participation, concept of impreciseness of modeling methods, basic requirements for end-user programming (e.g., critiquing systems, programming by example).
Intermediate level A framework being meta-designed in accordance with the concepts and beliefs of the meta level. It serves as an environment within which STSs are developed and do develop.	People (designers, managers, etc.) who are committed to meta-design will help to establish a framework within which various concrete socio-technical solutions can develop. This framework can include concrete software-developing tools, technical building blocks, modeling methods, organizational rules of participation, description of roles and tasks, and selection of personnel.	A KM environment established in a company to improve knowledge exchange by offering technical means and promoting appropriate social conventions. This environment can include a modeling method to specify process-oriented knowledge management. A CSCL environment as it might be introduced by a university's administration with which several concrete courses can be organized. A set of patterns of how concrete courses can be run may be included.
Basic level Socio-technical solutions as they are developed within the framework.	A concrete socio-technical solution as it exists during a certain period of time and will be a subject of continuous maintenance and adaptation.	A concrete document management system implemented to support a project. It includes categories of content and access rights; concrete rules and roles for its usage are specified. A concrete course for which students are assigned and instructed so that they can use the CSCL system.

existing system can provide for them. Meta-design *extends boundaries* by supporting users as active contributors who can transcend the functionality and content of existing systems. By facilitating these possibilities, *control* is distributed among all stakeholders in the design process (Fischer, 2007b).

Meta-design provides frameworks, which comprise objectives, techniques, representations of concepts, boundary objects, and processes for creating new media and environments that allow "owners of problems" as members of a social system to act as *designers*. A fundamental objective of meta-design is to create STSs that empower all relevant stakeholders of groups, communities of practice, communities of interest, and organizations to engage actively in the *continuous development* of a concrete socio-technical solution rather than being restricted to a prescribed way of interacting with the technical system or with its users.

The crucial aspect of meta-design, which leads to its name, is that of "*designing design*" (Fischer & Giaccardi, 2006). This refers to the concept of higher-order design, and the possibility of a malleability and modifiability of structures and processes as provided, supported, or influenced by computational media. It is a design approach that focuses on a framework of general structures and processes, rather than on fixed objects and contents.

Meta-design covers the *whole period of creative drafting of a solution*: specifying concrete concepts and plans (about technical infrastructure as well as organizational rules); introducing a technical system; experience with a first usage and feedback; the process of appropriation; and metamorphoses of the software system (Orlikowski, 1996) or the project goals (Herrmann & Hoffmann, 2005), including redesign. Therefore, meta-design is concerned with models of cyclic improvement and adaptation of socio-technical systems; these

models can comprise shorter and longer cycles of adaptation.

The higher-order concept of designing design becomes apparent by the three-level model of Table 2. The meta level contains the assumptions and orientation of how socio-technical meta-design should be organized as they are a matter of research; these are explained in the following sections. With these orientations, frameworks can be developed with which and within which concrete solutions can develop. These frameworks represent the intermediate level and combine technical and social issues to a socio-technical environment. On the basic level are the concrete socio-technical systems that develop or are developed with the help of such a framework.

Most powerful are those phenomena that serve as an example on all three levels. Wikipedia represents a very prominent example: on the basic level, it is a concrete solution for exchanging encyclopedic content; with respect to the intermediate level, it has emerged to a framework within which new tools are permanently adopted and social conventions assume increasingly more differentiated shapes; additionally, Wikipedia has inspired concepts on the meta level such as the belief that it is reasonable to support the role of *prosumers* in the web.

Meta-design can be characterized by the following five principles, which are discussed in detail and explained with concrete examples in the next section.

1. *Cultures of participation* (Fischer & Giaccardi, 2006) are concerned with the way in which designers and users can *collaborate* on the design activity, both at design time and at use time. Therefore, meta-design supports a culture of participation by which people with various and varying competences on the technical or domain level can contribute to shape a socio-technical solution. It puts *owners of problems in charge* and promotes a new distribution of control in socio-technical systems by establishing a culture of par-

ticipation. Methods and techniques of participatory design are provided for all kinds of stakeholders (e.g., end-users, managers, consultants, software developers, those who are responsible for quality management or privacy issues) to be involved. They all must have a chance to initiate the emergence of a socio-technical system or its appropriation and adaptation.

2. *Empowerment for adaptation and evolution*. The cultural and organizational framework being provided by cultures of participation has to be completed by specific methods and tools that especially empower end-users so that they can either partially take over the role of designers or can explain their needs to others who are able or have the right to adapt the features of a socio-technical system. End-users can benefit from critiquing methods and techniques (Fischer et al., 1998), from functionality for end-user programming, from descriptions explaining the rules and processes of a socio-technical system, from procedures of how others can be asked for help, from concrete examples of how a socio-technical system can be adopted, from all kinds of material with which they learn how to appropriate a socio-technical system, and so on. This kind of end-user support has to be provided by meta-design. For the context of socio-technical systems it has to be emphasized that end-users should be empowered not only to adapt the technical system but also to contribute to the development of social conventions, organizational rules, and definition of tasks, as well as other contributions.

3. *Seeding, evolutionary growth, reseeding*. The seeding, evolutionary growth, reseeding (SER) model is a typical principle of meta-design. Seeds or impulses can be represented by prototypes; by introducing new technology for a so-called pilot group within an organization; by an information campaign that prepares the implementation of a new system (e.g., KM); and by making people aware of their learning capabilities,

of needs for change, and of conflicts to be solved. If meta-design delivers concrete systems, these are meant only as examples and as seeds. They will always be accompanied with a frame of methods and tools that support development of these seeds and their evolutionary growth.

4. *Underdesign.* An important aspect of meta-design is *underdesign* (Fischer, 2003), which means that the structures and processes of an STS should be only partly specified; only those structures are determined that are indispensible to meet legal norms, security requirements, and basic economical needs. Therefore, it acknowledges the necessity to *differentiate between structurally important parts* for which extensive professional experience is required and therefore cannot easily be changed (such as structure-bearing walls in buildings) and *components users should be able to modify* to their needs because their personal knowledge is relevant (Habraken, 1972).

To support flexibility, underdesign includes examples of how things can be but need not be done; it provides maps instead of scripts (Schmidt, 1999), many options among which one can easily make a choice, and gaps to be filled in as well as guidance on how these gaps can be completed. This type of specification fulfills the need that everybody who is included can contribute to the completion of the design. It offers users (acting as designers at use time) *as many alternatives as possible*, avoiding irreversible commitments they cannot undo (one of the drawbacks of overdesign) (Simon, 1996). Underdesign is grounded in the need for *"loose fit"* in designing artifacts at design time so that unexpected uses of the artifact can be accommodated at use time (Henderson & Kyng, 1991); it does so by creating contexts and content-creation tools rather than focusing on content alone (Fischer & Giaccardi, 2006).

5. *Structuring of communication* for "*designing the in-between*" (Fischer & Giaccardi, 2006). Meta-design pursues the dual objective to support existing social networks and to shape new ones. It delivers methods of *appropriate communication support*—for example, strategies and methods for running participatory workshops, for facilitating discourses among stakeholders with differing perspectives (their needs and their ideas are collected and integrated), for enhancing social creativity, and for accompanying processes of the appropriation and adaptation of a certain technology. Meta-design aims to provide technology and methods that help to build social relationships, which mediates communication and supports negotiation among various perspectives. Promoting relationships among people includes affecting each other and being affected by social interaction. *"Methodologically, the third level of meta-design defines how co-evolutionary processes and co-creative, behaviors can be sustained and empowered on the basis of the way in which people relate"* (Fischer & Giaccardi, 2006). Both, artifacts as well as plans can serve as boundary objects (Star, 1989) that mediate the social interaction during design. Meta-design is concerned with the *identification and evolution of boundary objects* which help to connect the perspectives of a variety of stakeholders and to run as a thread through the whole life cycle from the idea of a new technology to its implementation into a socio-technical system and its appropriation. This life cycle can be methodologically accompanied by opportunities of facilitated discourses and reflections. A method of how such a discourse can be organized for the involved stakeholders is exemplarily outlined by the description of the socio-technical walkthrough at the end of the next section.

FIVE PRINCIPLES FOR META-DESIGNED STSS

Cultures of Participation

To support "designing together," meta-design facilitates cultures of participation that are different from the traditional *participatory design* (PD) approach (Kensing & Blomberg, 1998). Meta-design is based on the principles of PD, but it transcends them by taking into account new developments, such as (1) mass collaboration (Tapscott & Williams, 2006); (2) possibilities for end-user development (Lieberman et al., 2006; Pipek et al., 2009); and (3) agile software development (Cockburn & Highsmith, 2001; Fowler, 2001), in which customers and developers tightly collaborate.

The basic idea of PD is to allow all stakeholders to influence design-related decisions and give a voice specifically to those people who have in many case no influence because of imbalanced power structures; lack of knowledge, experience, or information; restricted communication capabilities; and/or technical reasons.

Meta-design transcends the traditional PD approach (Figures 1 and 2 illustrate the differences). Traditional PD usually aims at providing opportunities by which workers in a company can influence the design of tools that they will use afterwards to carry out their daily jobs. The relevant activities (from left to right in Figure 1) start with preparing and training stakeholders who will have to participate in decision making but are not used to doing so. These can be future users or their representatives, as displayed with the roles (ovals on the right side in Figure 1). They develop knowledge about the methods and tools (rectangle within the oval) which are used in the activity *participatory design*. This activity follows on "*preparing PD*" and employs typical PD-methods (left rectangle at the bottom). The phase of design is clearly separated (with a gray line in Figure 1) from the phase of the usage of the designed tools. In the case of traditional PD, design happens in workshops or meetings while employing the tools happens at the workplace; this is expressed with the activity "*work on regular, value-adding tasks*" in Figure 1. Traditional PD is grounded in a division of labor among managers, software engineers, and users. In this context, managers are in power on the social side, and engineers or developers are the power holders on the technical side (see role ovals in Figure 1).

By contrast, meta-design seeks to establish a culture of participation directly at the workplace combined with ongoing learning (see Figure 2) so that design can continue during the run time of a hardware/software system. Consequently, work on regular tasks and work on the employed infrastructure for these tasks are integrated. Meta-design promotes the quality that the set and the characteristics of the involved roles are highly dynamic: new roles emerge such as power users or co-developers (Nardi, 1993), and the traditional roles can continuously achieve and lose competencies that are needed to contribute to the development of their tools. Meta-design promotes a rich *ecology of participation* (Fischer et al., 2008; Preece & Shneiderman, 2009), which includes a broad variety of roles with varying characteristics, as shown in the elliptical symbol in Figure 2. The semi-circle in the role oval indicates that the list of roles is not complete. Meta-design tries to build a socio-technical environment (left rectangle at the bottom of Figure 2), which promotes the dynamic natures of roles.

Web 2.0 (O'Reilly, 2006) cultures are role models of how traditional roles (e.g., producers versus consumers) are dissolved and new roles, such as *prosumers* (Tapscott & Williams, 2006), are created. They demonstrate one of the essential strengths of cultures of participation: they have the potential to integrate a huge variety of different backgrounds, perspectives, and experience. The different roles are offered a variety of tools and activities, such as blogging, tagging, rating, and contributing.

Whereas traditional PD differentiates between clearly defined roles, meta-design aims on establishing a variety of roles and smooth

Figure 1. Traditional participatory design
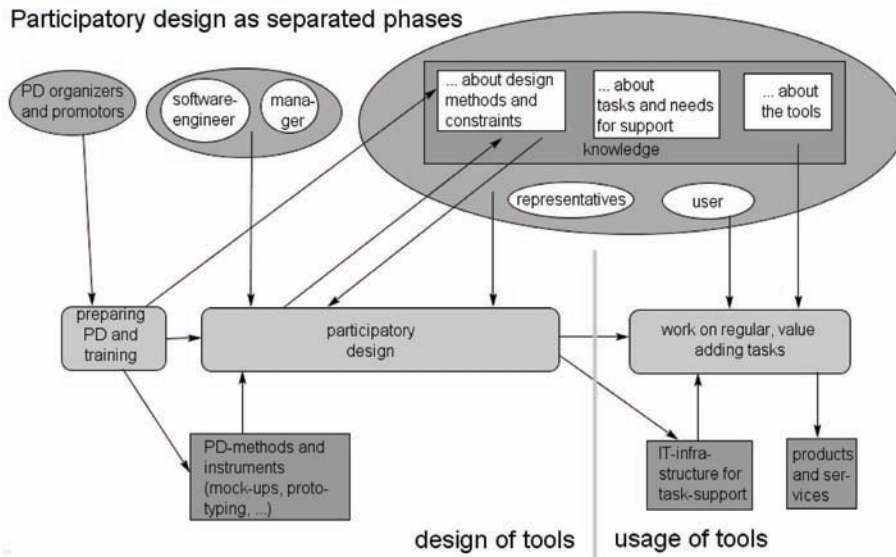
transitions among them. This includes shifting "*some control from designers to users and empowered users to create and contribute their own visions and objectives*" (Fischer, 2007b, p. 197). In the course of the evolution of an STS, developers and those who are originally responsible to maintain the system "*must accept a role in which they create mechanisms allowing users to act as designers and modify systems, thereby providing them with new levels of personal control*" (Fischer, 2007b, p. 202). This includes the fact that participation is not necessarily centrally organized; such a government can evolve if needed but is not a prerequisite of a culture of participation (Forte et al., 2009).

Table 3 represents the differences between traditional participatory design and establishing a culture of participation by meta-design.

People who are allowed or encouraged to participate are not always motivated to do so. Therefore, meta-design is also concerned with overcoming motivation barriers, with systems of rewards and incentives, and with promoting participation by methods such as facilitation or scaffolding. Users accept and exercise oppor-

tunities for participation only in the case of *personally meaningful problems* (Fischer, 2002). This paper mainly points out why the participation of various stakeholders in many roles leads to an improvement of STSs. However, this potential benefit is usually insufficient to motivate people to think continually in a design mode in addition to the other tasks in which they are involved. Deliberate research is needed to understand why and how people can develop the motivation to contribute to design instead on relying on fixed out-of-the-box solutions.

Empowerment for Adaptation and Evolution

Within socio-technical systems, users are not only those who directly interact with a technical system but all who benefit from the system as a whole when pursuing their interests or carrying out tasks. The permanent evolution of socio-technical systems is at least partially driven by their users, who share a wide range of possibilities for participation. Cultures of participation have to be complemented by tools

Figure 2. Cultures of participation—design in use
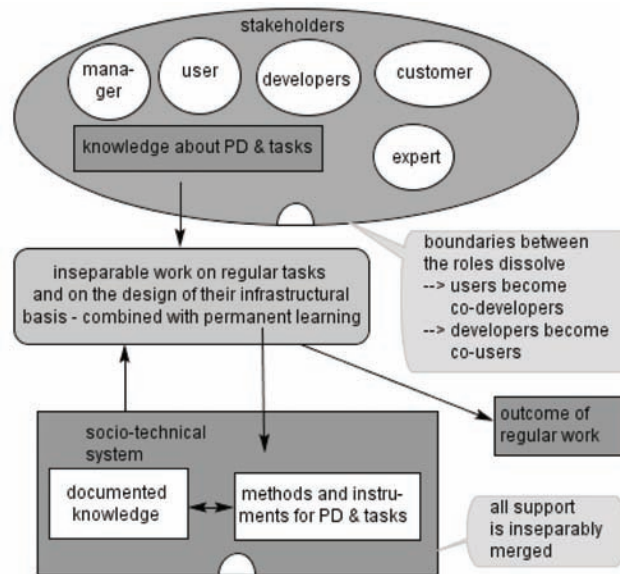
and methods that help users perform in the role of designers.

Adaptability of socio-technical systems by their users is different from the possibilities of *end-user development* (Lieberman et al., 2006). Even if the software system is almost not adaptable by a single end-user, it can become highly adaptable due to the self-adaptability of the socio-technical system as a whole. For example, the social system can develop and provide certain roles (e.g., support teams that can immediately react to the wishes of end-users if they need to modify their systems). Therefore, incremental improvement combined with intensive interaction with the users can take place. Meta-design of STSs is not focused on the software's adaptability by end-users (this is only one part of meta-designed features) but is concerned with the adaptability and means for the evolution of the STS as a whole. This includes possibilities to contribute to the evolution of organizational rules, social conventions, the culture of an organization, and so on. It is an important part of meta-design to differentiate among those cases for which:

- Software is directly adapted by end-users, either individually or in cooperation with other end-users;
- End-users closely collaborate with software developers, who immediately adapt the technical system; and
- Not (only) the software, but other structures or processes of the STS, are adapted.

As already pointed out in the previous section, meta-design aims at the evolution of an ecology of various and varying roles. These roles are also engaged in various ways and forms of collaboration in the adaptation of the STS. Therefore, a meta-designed framework has to provide a variety of tools, methods, processes, and strategies that supports all kinds of roles to take part in the adaptation and evolution of the various aspects of an STS.

Table 4 presents an overview of the aspects by which end-user development and meta-designed possibilities for the adaptation of the STS differ. It focuses on collaborative adaptation within socio-technical systems. Early studies (Nardi, 1993) already identified

Table 3. Participatory design and meta-design

	Participatory design	Culture of participation within a meta-design framework
Focus	Design time	Design *and* use time
Time line	The phase before the outcome of design is implemented; opportunities (e.g., workshops) are provided where participation takes place.	Design continues indefinitely, requiring active participation by users.
Tools and tasks	First, designing the tool; then carrying out tasks with the tool.	Working on the task and designing the tools needed for these tasks are intertwined.
Collaboration	The team that designs tools (technical infrastructure) and the team that collaboratively carries out the tasks with the technical infrastructure are separated.	The team that designs tools (technical infrastructure) and the team that collaboratively carries out the tasks with the technical infrastructure are overlapping or even inseparably merged.
Roles	Clearly separated roles such as workers, managers, developers, users, user advocates.	The boundaries between the roles dissolve, new roles emerge (co-developers, power users, prosumers), and the roles are highly dynamic.
Content	Information as content, on the one hand, and tools for information processing, on the other hand, are separated.	The development of the tool and the content are intertwined.
Application environments	Focused on work in companies with specific stakeholders, such as managers, developers, users.	Communities of interest and practice, open source communities, NGOs.
Regulations	Clear regulations about who is allowed to take part in decision making on what level.	Flexible degrees of involvement in decision making with the tendency to shift control from developers to users as co-developers.

that end-user development is more successful if supported by collaborative work practices rather than focusing on individuals. The studies observed the emergence of "gardeners" and "local developers" who are technically interested and sophisticated enough to perform system modifications that are needed by a community of users, but other end-users are not able or inclined to perform.

Seeding, Evolutionary Growth, and Reseeding Model

The SER model (Fischer & Ostwald, 2002) (see Figure 3) was developed as a descriptive and prescriptive model for creating software systems that best fit an emerging and evolving context. In the past, large and complex software systems were built as complete artifacts through the large efforts of a small number of people. Instead of attempting to build complete systems, the SER model advocates building seeds that change and grow, and can evolve over time through the small contributions of a large number of people. Therefore, these seeds play the role of *boundary objects* (Star, 1989), to which the communication between involved people can refer. SER postulates that systems that evolve over a sustained time span must continually alternate between periods of planned activity and unplanned evolution, and periods of deliberate (re)structuring and enhancement. It is apparent the the procedural model of SER also serves as guidance within meta-designed frameworks for the development and evolution of socio-technical systems. In STSs, seeds need to be available for the technical components as well as the social structures and processes.

The SER model encourages system designers to conceptualize their activity as meta-de-

Table 4. End-user development and usage-oriented development and adaptation in STSs

End-user development	Usage-oriented development and adaptation in STSs
Adaptation mainly by programming, parameterization, configuration, etc.	Adaptation by communication in the course of incremental cycles of demand—getting it programmed, testing it, new demand—with minor parts of programming by the user.
Mainly individual development with some collaboration between end-users and involvement of experts.	Collaborative developing is shared among various roles.
Individual learning by the end-user.	Collective learning of people in various roles of the socio-technical environment.
Gentle slopes of increasing complexity.	Gentle slopes of involving more and more parts of the socio-technical environment.
The user interface is decisive to make end-user development possible.	The interfaces to others is decisive, to make communication for cycles of agile development possible.
The system shows the end-user how its features can be modified.	Others show end-users how they can modify their systems.
The offered functionality mainly aims on the adaptation of software.	The adaptation refers to technical as well as social, and organizational structures and processes of carrying out tasks, learning, etc.

Figure 3. The seeding, evolutionary growth, and reseeding (SER) model
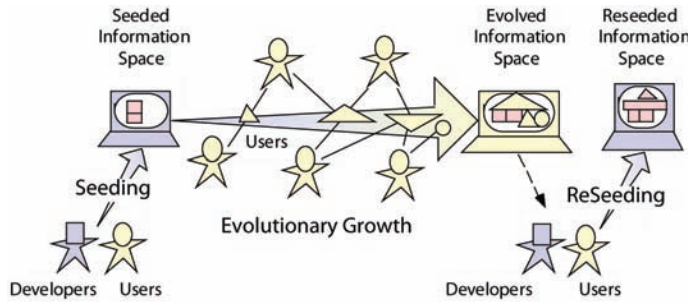
sign, thereby aiming to support users as active contributors. The feasibility and usefulness of the SER model for reflective communities has become apparent in the context of several areas (see the next section).

Meta-design provides methods and practices that support seeding and evolutionary growth. SER works only in the context of the other principles of meta-design such as participation, underdesign, and empowerment for adaptation. Similar to action research (Avison et al., 1999) or the behavior of reflective practitioners (Schön, 1983), phases of experimenting and practicing have to alternate with phases of reflection during the evolutionary growth. Transferring the SER model to STSs implies that seeds are built not only for technical features but also for social structures and interactions. The growth of the seeds (for both the technical and social dimensions) cannot be anticipated at design time. How seeds will evolve or are

used is situated in future uses at use time and cannot be sufficiently planned at design time.

Underdesign of Models of Socio-Technical Processes

Underdesign can refer to either concrete artifacts or plans of how the artifacts should be designed. It can also refer to either how the design project will be organized or how the usage of the artifact is coordinated among several people for collaborative tasks. A subset of these plans may be represented by graphical models for software-design (e.g., with the unified modeling language, or UML) or for process management; others may be checklists, Gantt charts and so on. The modeling method SeeMe represents a special approach with which flexible degrees of under design can be chosen by varying the degree of completeness and preciseness.

As previously pointed out in the subsection, "A Conceptual Framework for Meta-Design," underdesign in the context of STS not only refers to hardware and software but also to the plans that describe how the technology will be used and how the collaboration of the users is coordinated. The most prominent examples of representing this kind of plan are process models. They can be overdesigned, as in the case of models that are developed to program workflow management engines. Preprogrammed workflow management systems force the users into inflexibility, which presents problems in handling exceptions or improvising a solution, for example (Thoresen, 1997). Conversely, it is not reasonable to go without explicit process models (Schmidt, 1999) because they help people within an STS explain the need for changes to others, introduce newcomers to the STS, or document changes that have taken place so that evolutionary growth is supported. The solution is a modeling method incorporating underdesign with flexible degrees of incompleteness and impreciseness.

The modeling method SeeMe (semi-structured, socio-technical modeling method) has been developed to represent concepts and processes of socio-technical systems and also to articulate incompleteness, uncertainty, informalities, and freedom of decision. Therefore, SeeMe offers the possibility to represent vagueness explicitly and to choose flexible degrees of underdesign (Goedicke & Herrmann, 2008). The method aims to the integration of technical and social aspects as well as formal and informal structures. Therefore, it visualizes the complex interdependencies among different people, between humans and computers, and among technical components.

The concept of SeeMe and examples of its usage have been described in several papers (Herrmann & Loser, 1999). Therefore, the following explanation focuses on the relationship between SeeMe and underdesign. The model in Figure 4 represents the basic concepts of SeeMe by displaying a real example from a KM project of a manufacturing company that produces electric control boxes for the mining industry. Within this context, the diagram displayed in Figure 4 is a concrete example of an initial seed that had structured the discussion about a KM system and helped to evolve the descriptions of the needs and requirements that were assigned to the new system (see the subsection "Knowledge Management" later in this paper). The diagram in Figure 4 contains the three basic elements of SeeMe: roles, such as "mechanical worker"; activities, such as "mechanical work on electric control boxes" or "preparing and planning"; and entities such as "electric control box components." These elements can be embedded into each other. Relations are represented with arrows, which express that one activity is followed by another, that roles carry out activities, or that entities are used or modified, among other relations.

Figure 4 focuses on the tasks of the role of "mechanical worker" but shows them in the context of other roles. On the left side of the diagram, the already available tools are displayed, and at the bottom, the components of the KM system are only roughly outlined. It turned out that focus on KM needs to consider the administrative tasks in more detail. Therefore, the activity "mechanical work on electric control boxes" is represented with two perspec-

Figure 4. Knowledge management in the context of manufacturing
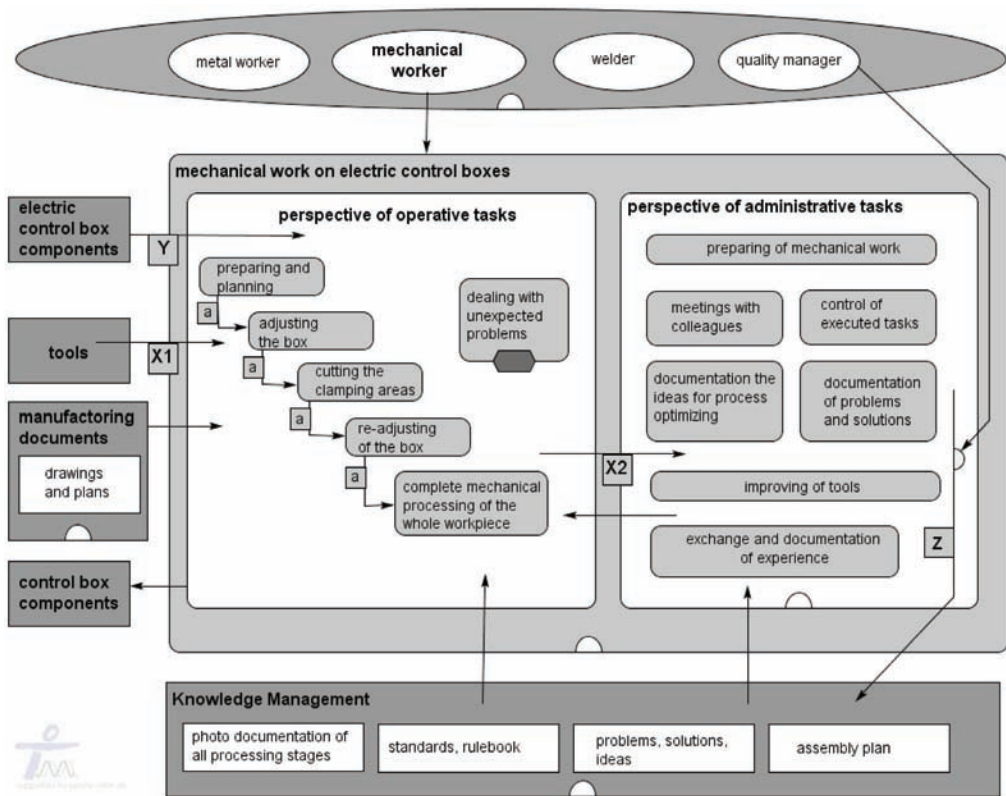
tives, operative versus administrative. Figure 4 includes a specific relation that points from "quality manager" to relation "Z." This specific relation expresses that the quality manager is interested in activities that lead to entering information into the KM system. Interests are a typical phenomenon that characterizes social interactions.

Figure 4 shows some central examples for methodological aspects to support underdesign:

- **Incomplete specification of subelements**. The listed roles, the manufacturing documents, the administrative tasks, and the KM components are only incompletely specified. This is expressed by semicircles. For instance, in the case of the administrative tasks, the involved discussants were not sure whether they had mentioned all

the important activities. By contrast, the operative tasks have been considered as the only example where an activity can be completely specified because the mechanical work appeared as well specified due to the clear definition of the outcome that had to be achieved.

- **Freely sequenced and overlapping activities versus determined sequence of activities**. A further contrast between these two perspectives refers to the sequencing of activities. The relations of type (a) in the operative tasks activity expresses that the displayed sub-activities (preparing, adjusting the box, etc.) are strictly sequenced, whereas such a sequencing is not obvious for the adminstrative tasks. The graphical concept of embedding activities (Harel, 1987) helps to express that the employees

can freely decide by themselves how the activities are sequenced: whether they want to proceed in a certain sequence, whether this sequence changes from case to case, or whether they work simultaneously on some of the included subactivities. If it turns out after a while that it is reasonable to carry out some activities in a prespecified sequence, the model could be changed afterward and the knowledge management could be adapted to support this sequence. If a sequence of administrative tasks were sequenced at the beginning of the project, this would have been an example of overdesign.

- **Predetermined decisions versus freedom of decision**. The activity "dealing with unexpected problems" is annotated with a hexagon, which usually expresses that this activity takes place only under certain conditions. However, the hexagon is empty in this context—the conditions under which a problem is considered as exceptional (e.g., the customer requires changes after the beginning of the production) are not explicitly listed. Subsequently, the employees decide whether they consider a problem as exceptional or as routine.

- **Unspecified transitions and relations**. The relation labeled X1 cuts into the entity of "tools." This means that only a subset (not all) of the tools are used, and that it is not reasonable to specify this subset in advance. Therefore, the cutting arrows are another possibility for underdesign. Similarly, the relation X2 expresses that it is not appropriate to specify the operative subtasks from which administrative tasks are exactly initiated. Therefore, this specification is left to the workers when they start to document the handling of a case. The administrative tasks can start before the operative tasks are completed. Such a constellation is typical for everyday work practice—one manager has described this configuration as "diagonally parallel" activities. By contrast, the left side of the relation labeled Y expresses that all the components of the electrical box have to be objects of the mechanical work since it is not cutting into this entity.

- **Meta-relations**. Beside what is displayed in Figure 4, SeeMe offers the possibility of a meta-relation that helps to express that the diagram includes activities or roles that are able to change the structures currently represented in the diagram. The meta-relation has a self-referential meaning and is closely related to the intentions of meta-design. The meta-relation usually points from activities or roles to the structures that can be modified. For example, meta-relation can be used to express that a project manager determines which roles or persons will participate in a project team.

There are, in principle, two possibilities to deal with incompleteness, which are indicated in SeeMe diagrams: it can either be eliminated and replaced by more complete specifications in the course of design and usage, or the incompleteness remains and opens a space for free decisions that are "taken on the fly" and depend on the context where, for example, a software system is used. It has to be emphasized that even if parts of a diagram are completely specified this does not necessarily imply that the real processes will run exactly as specified. The models are only a first approach to understand or to plan what happens in reality, and they have to be negotiated and adapted continuously. SeeMe is not the only modeling method to document the planning of socio-technical processes. Others also pursue this purpose, but only few support explicitly dealing with vagueness, such as i* (Yu & Mylopoulos, 1994), which differentiates between hard goals and soft goals identified during the requirements analysis.

Structuring of Communication

The modeling method SeeMe supports design on the level of planning. Whether and how the specifications of a plan are brought into reality is by no means determined by the plan itself, but depends on communication processes and

how the people within a socio-technical environment are related to each other. So although software can be programmed and configured, the implementation of new organizational structures and processes is a matter of complex communication.

Meta-design can help to support this communication by certain interventions, such as bringing people together by organizing workshops and facilitating them. We propose a method called the *socio-technical walkthrough* (STWT) (Herrmann, Kunau, Loser, & Menold, 2004), which has matured in the course of several cases (Herrmann, 2009). The STWT consists of a series of workshops. In every workshop, a model of the STSs—such as SeeMe diagrams—is discussed, completed, and negotiated. The facilitation of these discourses is walkthrough-oriented: "structured walkthrough" (Yourdon, 1979); "cognitive walkthrough" (Polson et al., 1992); or "groupware walkthrough" (Pinelle & Gutwin, 2002). The STWT can be characterized by its facilitation strategy:

*   *Getting started*: The facilitator usually prepares a diagram representing the plan of a STS. It is reasonable to begin with an overview diagram and to have a strategy of how to walk through the diagram step-by-step.
*   *Asking prepared questions*: With every step, the facilitator focuses on parts of the diagram and, for every step, applies one or two prepared questions, such as: "Which kind of information is needed or produced here?" or "How can the information processing be technically supported?" The stakeholders are encouraged to respond to these questions.
*   *Collecting contributions*: The facilitator collects the answers, hints, proposals, comments, references to further documents, and so forth. It is important that the stakeholders contribute their varying, and potentially conflicting, viewpoints and make comments.
*   *Focusing on the diagram*: The diagram serves as a "boundary object" (Star, 1989),

which integrates the varying perspectives of the participants into a larger picture. Therefore, the facilitator makes sure that the collected contributions are inserted into the diagram. The diagram's growth mirrors the ongoing discourse. Everybody's contributions are valued and must leave traces in the diagram. This does not necessarily imply that every proposal shapes the outcome of the design, only that it has a chance to do so.

*   *Dealing with conflicts*: making differing *positions comparable* and visible helps to deal with conflicts and to "*support congruence*" (Cherns, 1987, p. 158*)*. Depending on the social context, the eventual solution to a conflict is found by negotiation or by a decision of the management. These decisions can also be postponed until the first practical experience with the socio-technical solution has been made.

Between the workshops, the resulting diagrams can be discussed with others who have not participated in the workshop, they can be compared with the reality of everyday practice, they can be reconsidered by experts, and their appearance can be improved to increase their comprehensibility.

Therefore, the STWT is a method to support participation and to give users the opportunity to decide how a technology will be shaped and collaboratively used. The STWT offers users possibilities for permanent learning and a means to express themselves so that they can document their ideas and demands for adaptation, communicate them to others, learn how to bring them into reality (by themselves or with the help of others), and finally check whether the outcome of adaptation complies with their goals. The diagrams and the technical artifacts to which they refer can be considered as seeds; the STWT workshops provide a place where the evolutionary growth of these seeds can take place (with respect to the diagrams) or be reflected (with respect to the technological change that is mirrored in the diagrams).

SeeMe diagrams are only one example of the type of artifacts that can be used for the STWT. Other kinds of artifacts may be scenarios (Carroll, 1995), UML-based use case descriptions, or presentations of personas (Grudin & Pruitt, 2002). The indispensable characteristics are that they can be inspected step-by-step, that they support underdesign, and that they serve as boundary objects that can be understood and shaped from the background of various perspectives and therefore serve as a seed for the evolution of an STS. A STWT is usually centrally organized by a facilitator. However, within a culture of participation, the role of the facilitator can be taken by varying stakeholders.

EXAMPLES OF META-DESIGNED STSS

Relevance of Meta-Design for a Broad Spectrum of Applications

Meta-design provides *conceptual frameworks* (e.g., contexts for creating content; see the subsection "Cultures of Participation"), *processes* (such as the SER model; see the subsection "Seeding, Evolutionary Growth, and Reseeding Model"), and *tools* (such as SeeMee; see the subsection "Underdesign of Models of Socio-Technical Processes"). It provides a fundamentally different *design methodology* for a broad spectrum of application areas, including:

- *Software design*, with a focus on customization (Henderson & Kyng, 1991); personalization, tailorability (Mørch, 1997); design for diversity (Carmien & Fischer, 2008); and end-user development (Lieberman et al., 2006);
- *Architectural design*, with a focus on underdesign (Brand, 1995; Habraken, 1972);
- *Urban planning*, with a focus on land use, public transportation, and flood mitigation (Fischer, 2006) as pursued by the Envisionment and Discovery Collaboratory (EDC; see the discussion later in this section);

- *Teaching and learning*, with a focus on learning communities (Rogoff et al., 1998), courses-as-seeds (dePaula et al., 2001), and negotiation of concepts (Carell & Herrmann, 2009; Herrmann, 2003);
- *Living information repositories*, with a focus on organizational memories (dePaula, 2004) and community digital libraries (Wright et al., 2002);
- *Interactive art*, with a focus on co-creation by putting the tools rather than the objects of design in the hands of users (Giaccardi, 2004);
- *Web2.0-based cultures of participation*, with a focus on informed participation (Brown et al., 1994); collaboratively constructed artifacts (Scharff, 2002); and social creativity (Fischer, 2007a); and
- *Knowledge management*, with a focus on bottom-up–oriented knowledge contribution (Diefenbruch et al., 2000; Herrmann et al., 2003a) (see the discussion later in this section).

In the following subsections, two types of frameworks (collaboratories and KM) that have a twofold character with respect to meta-design are described in more detail. These are socio-technical systems that are meta-designed and are frameworks where design takes place.

Collaboratories

A *collaboratory* (Finholt & Olson, 1997) is a place where people come together to work on such tasks as design, planning, developing visions, and solving concrete problems, and are willing to collaborate, to learn from each other, and to permanently reflect and improve the tools and methods they use. The constituents of a collaboratory are not only the technical infrastructure; they also include:

- People who dynamically share various roles and tasks as well as their social interaction; they are users of the collaboratory;

- Places where results are documented and archived;
- Properties of the collaboratory, such as subjects of reflection and making proposals for improvement; and
- Some people who prepare sessions in the collaboratory and maintain it, some who have the task to develop visions of how the collaboratory can evolve, and some who work on adapting the technology and contributing to incremental improvement.

Collaboratories are places where heterogeneous perspectives are melted, transdisciplinary cooperation takes place, and learning is continuously going on. They are special but typical examples of STSs, and their properties and constellation are very flexible and include a wide range of possibilities for further development so that they can be considered as the typical outcome of meta-design. This can be outlined by the concrete examples of two collaboratories, ModLab and the Envisionment and Discovery Collaboratory (EDC).

ModLab: A Facilitation Collaboratory

The ModLab was developed to facilitate design-oriented communication among various stakeholders and to support collaborative creativity (Herrmann, 2010). Its centerpiece is a large, high-resolution interactive wall (4.80 m × 1.20 m; 4,320 × 1,050 pixels, which seamlessly integrates three rear-projection boards (see Figure 5). Touches are recognized via six cameras that view the reflection of infrared light caused by fingers or pens. The angles of view of the cameras overlap to support uninterrupted dragging actions over the entire wall. Data can be entered and manipulated directly on the screen or via laptops connected via WiFi. At the moment, mainly three types of software are available: the Microsoft™ Office suite; an editor for process diagrams (www.seeme-imtm. de); and the SMART™ software, which is used to control the interaction with the board but also provides means for notetaking, handwriting

recognition, annotations on PowerPoints, and so forth. Furthermore, we identified some web applications (e.g., Google Docs, Mindmeister) that support collaboration within and between meetings. This collaboratory is frequently used to run workshops where brainstorming is conducted or socio-technical processes are designed. Recent examples include a workshop on the development of tagging mechanisms for process models (Prilla, 2009) and a meeting for identifying useful services that can be offered to elderly people (Carell & Herrmann, 2010).

The project leaders who organize the meetings in the collaboratory continuously try to find new tools that can be used in the lab, ask other people who are responsible for the maintenance of the lab to install these tools, and test them. Users who visit the lab have to get used to the new types of technologies, develop preferences and reservations, and make proposals for improvement.

The Envisionment and Discovery Collaboratory (EDC)

The EDC (Arias et al., 2000) is a long-term research platform that explores conceptual frameworks for new paradigms of learning in the context of design problems. It represents a STS supporting reflective communities by incorporating a number of innovative technologies, including table-top computing environments, the integration of physical and computational components supporting new interaction techniques, the support of reflection-in-action as a problem-solving approach (Schön, 1983) and an open architecture supporting meta-design activities.

The EDC brings together participants from different domains who have different knowledge and different contributions from various backgrounds to collaborate in resolving design problems. The contexts explored in the EDC (e.g., urban planning, emergency management, and building design) are all examples of ill-defined, open-ended design problems (Rittel & Webber, 1973).

The EDC serves as an immersive social context in which a community of stakeholders

Figure 5. ModLab—a facilitation collaboratory at the University of Bochum

can create, integrate, and disseminate information relevant to their lives and the problems they face. The exchange of information is encouraged by providing stakeholders with tools to express their own opinions, requiring an open system that evolves by accommodating new information. The information is presented and handled in a way that it can be used as boundary objects. For example, city planners contribute 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. Figure 6 shows the EDC in use, illustrating the following features.

- The pane at the bottom shows a table-top computing environment that serves as the action space: the stakeholders engage in determining land use patterns as a collective design activity in the context of an urban planning problem.
- The left pane at the top is the associated reflection space in which quantitative data (derived dynamically from the design moves in the action space).

- The right pane at the top visualizes the impact of the height of new buildings (sketched by the stakeholders in the action space) on the environment by using Google Earth.

We have begun to include mechanisms within the EDC to allow participants to inject content into the simulations and adapt the environment to new scenarios. The next steps include creating ways to link to existing data and tools so that participants can draw on information from their own areas of expertise to contribute to the emerging, shared model. By exploring these different approaches, the EDC has given us insights into collaboration that draws on both individual and social aspects of creativity.

A META-DESIGN PERSPECTIVE ON THE COLLABORATORIES

Both ModLab and the EDC are specific examples of STSs that have a number of characteristics in common. These commonalities illustrate the following aspects of meta-design:

Figure 6. The Envisionment and Discovery Collaboratory (EDC)

Technical Infrastructures and Social Interactions of Various Roles are Intertwined

Bringing the technical infrastructure of the collaboratories into existence was constitutive for the development of a community that integrated technicians, researchers, and users of the collaboratories. After such a community had evolved, it started to make design proposals for enhancing the collaboratory's technical components. In this way, the technical infrastructure and its community of users, technicians, and researchers (with a core group of about 10 people) evolved itself as a socio-technical unit.

Most of the new technical features that were implemented in a collaboratory (e.g., the usage of gestures on the interactive wall; the activation of commands by positioning objects on the table top) didn't work very reliably in their starting phase as prototypes. The reactions of these new features appeared as contingent with respect to the input actions of the users.

Therefore, a phase of maturing was triggered by the technicians to eliminate contingent reactions (cf. Table 1; control), and to make HCI sufficiently reliable. The more reliably it worked, the more the community was able to let new ideas emerge, which inspired the ongoing design of the collaboratory's infrastructure (e.g., developing a game that helps newcomers become familiar with the technological support). Those types of contingency that were based on technical malfunction motivated the technical staff to eliminate them, and they also were inspiring the users to develop new ideas. The collaboratories are a place where people start to "play around"—either in reality or in their imagination—with the available features. This was also a source for inspiration (cf. Table 1, situatedness, contingency). Actually, the collaboratories were not built to continue the design of their own infrastructures but to support design in other areas, such as urban planning or service engineering. However, working in these design areas did incidentally

contribute inspirations for the improvement of the collaboratories themselves.

Collaboratories Evolve in Cultures of Participation with a Variety of Participants in Various Roles

Whereas traditional PD (see Table 3) would have emphasized the phases of drafting and planning of a collaboratory such as the Modlab, the main participation of the collaboratory at the University of Bochum started only after it was established. A collaboratory is such a complex phenomenon that it is difficult to imagine its possibilities before its features and potential are experienced by being inside. According to Table 3, the phase of usage itself was most important for the development of a culture of participation (see the subsection "Cultures of Participation"). There is also no definite point of time when the design of a collaboratory's infrastructure comes to an end; in contrast, it seems to go on indefinitely (see Table 3, time line). An ecology of roles (see Table 3, roles) has evolved during the evolution of collaboratories (Fischer et al., 2008), such as:

- A *project leader*, who is responsible for the overall design and the usage of the collaboratory;
- One or more *chief technicians*, who solve technical problems and evolve the infrastructure;
- *Personnel (e.g., students)*, who maintain the hardware and software to develop new features;
- *Domain experts*, who solve problems of their domain with the help of the collaboratory;
- *Scientists*, who use the collaboratory as members of research teams;
- *Students and teachers*, who use the collaboratory for learning and knowledge construction; and
- *Typical test-persons*, who detect every problem with a technical feature by their experimental usage behavior.

In the case of a traditional PD, one would have tried to clearly define the competencies of the involved roles so that their responsibility and authority can be made visible for all participants. By contrast, in the case of an evolving culture of participation, the tasks, activities, and competences of these roles can overlap: The technical infrastructure can be considered as a domain itself, and problems of this domain are discussed and partially solved by everybody in the collaboratories; the experts of other domains can contribute with proposals for technical improvement; and users become co-developers and developers become co-users. Users start to observe the troubleshooting routines of the technicians and begin to solve little technical problems by themselves. Teams of technicians and users cooperate very closely (see Table 3; collaboration). The social system as a component of the socio-technical collaboratory continuously evolved. This was also triggered by the integration of new personnel, who contributed new perspectives and knowledge domains.

Adaptation of the Technical Infrastructure Is User-Driven

The technical infrastructure has been continuously adapted to the needs of the people. This did not happen mainly by employing mechanisms of end-user programming. In contrast (cf. Table 4), the users either delegated certain tasks (mainly adding new features to the collaboratory) to the technicians, and the technicians explained how the users could handle technical problems by themselves.

A typical example is the calibration of the touch screen in the ModLab. Adaptations are carried out or promoted by those who maintain the collaboratory (see the subsection "Empowerment for Adaptation and Evolution"). The users develop new ideas of how they can convey or present their information and they start by trying out various possibilities of new information exchange; this inspires them to ask the technical staff to provide them with new features (such as wii-controlled interaction,

touch-based rating mechanism, etc.). Once again, these proposals inspire the technicians to develop and implement their own ideas for improvement. Mutual learning and collaboration are the bases for the ongoing adaptation and maintenance where people increase their availability to take over the viewpoints of others.

In the course of this collaboration, not only technical infrastructure was adapted but also the social system, for example, by integrating new people into the staff who maintain the collaboratory. These newcomers brought in new perspectives and ideas of how the collaboratory could be enhanced and used. An important prerequisite for the continuous development of a collaboratory is to design it as an assembly of building blocks or components that can be flexibly and experimentally combined (Mørch et al., 2004). Examples for these building blocks are software features, web applications, and hardware devices, among others.

From the perspective of meta-design, collaboratories are self-referential socio-technical systems: they are designed to evolve, they are the place where this evolution takes place, they provide the infrastructure that supports this evolution, and they provide the context that represents the common ground on which this evolution is driven by the communication between problem owners.

Knowledge Management

KM has a twofold character in the context of socio-technical meta-design: on the one hand, STSs are designed to support knowledge exchange, and on the other hand, knowledge exchange and integration (Herrmann et al., 2007) are needed in the course of the development of an STS. KM strategies have developed in companies that pursued the goal to be aware of the firm's knowledge resources, to continuously evolve them, and to make them mutually accessible (& Leidner, 2001). Therefore, technical systems were employed to store the knowledge and to distribute it. Additionally, it was intended to integrate the various sources and repositories of electronic documents. Strategies of KM

are also applied for the knowledge exchange between companies and within communities. Web2.0 paradigms (O'Reilly, 2006), especially the emergence of Wikipedia, had a tremendous influence on KM-strategies in firms where one attempts to copy the success of bottom-up–oriented knowledge exchange and users are empowered to contribute and adapt content (see the subsection "Empowerment for Adaptation and Evolution"). Wikipedia is an example of how people who don't have an official status as experts in an certain area can contribute to an encyclopedia, and it demonstrates mutual collaboration where expert status and power relations have at least secondary relevance (Benkler, 2006).

In many cases, KM projects tried to develop and introduce a concrete technical system, for example, BSCW (Appelt, 1999) and Answer Garden (Ackerman, 1998), to support KM for a certain purpose, such as project management or support of a hotline, and certain conventions, such as how, when, and where documents have to be stored. This kind of socio-technical solution represents an STS. However, this solution never stands alone but has to work in the context of other systems that are used for KM activities and have either been developed systematically or emerged in the wild. In companies as well as on the web, there is not just one type of system or application which supports knowledge exchange and not only one type of behavior for distributing and integrating knowledge, but a whole variety of them that build a socio-technical framework (see Table 2, intermediate level). It can be considered a task of meta-design to provide such a framework where concrete solutions can develop that cover:

- Plans and strategies of how knowledge exchange can be improved;
- Technical applications that are used to collect, structure, and distribute knowledge;
- Processes and conventions of how knowledge will be documented and used;
- Content representing the relevant knowledge;

- Support of learning in the context of knowledge construction and knowledge application; and
- Meta-knowledge that represents information about the value of knowledge, how it is structured and used, etc. (Herrmann, Kienle, & Reiband, 2003).

A meta-designed KM framework is an STS in which various roles collaborate in a culture of participation (see the subsection on cultures of participation), and concrete plans, technical features, commitments, and so forth can be considered as seeds (see the subsection on the SER model) that are adapted step-by-step and help to evolve and initiate new habits of knowledge exchange.

A SeeMe diagram similar to that shown in Figure 4 was developed at the start of a KM project for a manufacturing company and served as an initial, underdesigned plan (see the subsection "Underdesign of Models and Socio-Technical Processes")—a seed that grew over a period of six STWT workshops (see the subsection "Structuring of Communication"). The final result contains about six times more elements than that shown in Figure 4. The roles displayed in Figure 4 as well as a project leader were involved in the STWT. The content of the KM and the first experiences with it were discussed in the workshops, which were also used to train the usage of the system and to initiate organizational change. The most relevant aspect of the project was that the participation of the workers has been introduced as a sustainable element of continuing reflection and continuous improvement—this can be interpreted as an initiation of a culture of participation. The discussion about what the KM system should offer already helped them to develop a better understanding of their own work and their collaboration.

Another example deepens our considerations on the relationship between meta-design and cultures of participation. We helped to introduce a KM solution for central consumer counseling in North Rhine Westphalia, Germany, which supports more than 50 local advice centers (Herrmann, Hoffmann, Kunau, & Loser, 2004, p. 18). The basic idea behind the project was to provide and distribute information needed to help people to make their decisions when they buy products or services. They can also seek the help of professional counselors for these decisions. Their work has to be supported by the KM project. A system was introduced that provided documents and the latest news about products and services available on the German market. Due to legal reasons, the information flow was only in one direction: from the central organization to the local advice centers. The central organization was legally responsible to make sure that the distributed information was correct. Therefore, local agents were not allowed to enter information into the system, although they gained a lot of experience and would have preferred to document these data in the KM system. Therefore, the motivation to work with the system was not very high—paper-based documents, to which additional information could easily be annotated (e.g., with post-its), were still more favored three years after the system's introduction. Furthermore, the one-directional flow of information was even fixed by the type of technology itself because the central organization had purchased only a few software licenses to allow the users to enter information; most of the licenses were valid only for a read-only access. It is apparent that a meta-design approach could have helped to overcome some of these problems, as outlined below.

- It would have promoted a much more flexible technical solution by which the access rights could have been flexibly adapted.
- A continuous process of negotiation and adaptation would have been implemented whereby the conflicting needs of adding personal information, distributing it, and delivering legally secured information could have led to a solution that presented appropriate compromises; the quality ensuring and rewarding procedures of Wikipedia could serve as a role model in such a case (Bryant et al., 2005).

- The continuous learning by the employees about what is possible with the system would have accompanied the continuous process of adaptation.

GUIDELINES FOR THE META-DESIGN OF STSS

This section describes *guidelines* (Fischer et al., 2009) derived from our conceptual considerations (see the sections on meta-design and practical experiences) with the development of STSs. These guidelines transcend the principles and propositions for socio-technical design as proposed by Cherns (1987) and Eason (1988):

- "Principle 1: Compatibility … Members must reveal their assumptions and reach decisions by consensus … Experts are needed … they, too, are required to reveal their assumptions for challenge" (Cherns, 1987, p. 154f).
- "Principle 2: Minimal Critical Specification. … no more should be specified than is absolutely essential … requires that we identify what is essential" (Cherns, 1987, p. 155).
- "Proposition 3: The effective exploitation of socio-technical systems depends upon the adoption of a planned process of change" and
- "Proposition 9: The exploitation of the capabilities of information technology can only be achieved by a progressive planned form of evolutionary growth" (Eason, 1988, p. 46f).
- "Proposition 6: The specification of a new socio-technical system must include the definition of a social system which enables people in work roles to co-operate effectively" (Eason, 1988, p. 47).

These principles and propositions suggest that mainly needs an actor is necessary (a manager or designer) who can recognize a certain principle (e.g., that members reveal their assumptions), outline a plan (how things should evolve), or define a social system as it should

be—and this is sufficient to bring a successful socio-technical solution into reality. By contrast, meta-design aims to provide the basis on which STSs can develop with respect to the goals that are behind the above-quoted principles.

Provide Building Blocks

From a technical point of view, a meta-design framework should include components and building blocks for HCI, software functionality, and content. These are hardware devices, software features, documents, presentations, web applications, web sites, etc. as they are used in STSs, such as the described collaboratories and knowledge management solutions. The users of an STS can freely combine, customize, and improve these components or ask others to do so (see the subsection "Empowerment for Adaptation and Evolution"). It is not reasonable to provide a complete, integrated set of components as a final technical solution to which a social system should adapt. By contrast, the meta-designed framework may include only complex technical solutions if they are meant as examples of how the components can be integrated, but not as prescriptions. These examples should have the role only of seeds, which inspire the evolutionary growth of a new assembly of components that fits into the STS. Meta-design must be continuously aware of new technological trends, and the meta-designed framework must be flexible enough to integrate these trends by providing new building blocks. They must be suitable as seeds that give impulses for new directions of evolutionary growth (see the subsection on the SER model) in concert with the already existing components.

Underdesign for Emergent Behavior

Systems need to be *underdesigned* so that they are viewed as *continuous beta* that are open to facilitate and incorporate emergent design behaviors during use. Underdesign is not less design but more design because a meta-designed framework provides meta-tools, meta-methods, and meta-knowledge to allow

people with various and varying competences to collaboratively design socio-technical solutions. A meta-designed framework establishes a corridor within which participatory design can develop without re-inventing the wheel or violating such constraints as legal norms, ethical restrictions, and the like. Underdesign helps to answer the question of how complex the technical building blocks that are provided by a meta-designed framework should be: On the one hand, they should integrate enough functionality so that a useful and reasonably usable unit is offered. On the other hand, they should not be too complex or they would have to be "disassembled" if someone wants to combine them with other building blocks. Underdesign has also to be applied to planning. In contrast to Eason's propositions, we do not assume that the evolution and change within STSs can be fairly planned. Therefore, methods of documentation have to be employed for the planning of an STS that allows for incompleteness and impreciseness (see the subsection on underdesign). Plans are meant as seeds (see the subsection on the SER model). They neither completely describe what should or will be nor do they completely match all aspects of the reality of an STS.

Establish Cultures of Participation

People should be enabled and attracted to bring their competences and perspectives into the development of socio-technical systems. Therefore, a transparent policy and procedure is needed to incorporate user contributions. To attract more users to become developers, the meta-designed framework must offer "gentle slopes" (see Table 3) of progressive difficulty and incremental extension of the included design aspects so that newcomers can start to participate peripherally and move on gradually to take charge of more difficult tasks. Important relevance has the structuring and facilitation of communication (e.g., by walkthrough orientation; see the subsection "Structuring and Communication") so that all kind of participants are encouraged to make their contributions and can realize that these

contributions are recognized and become part of the decision-making process. Rewarding and recognizing contributions is an essential prerequisite of fostering intrinsic motivation. Roles and their rights and duties must not be fixed for the period of an STS's evolution but should be part of this evolution so that domain experts can become co-designers, new roles can be integrated and control can be shifted in accordance with increased competencies (see the subsection "Cultures of Participation").

Share Control

A further crucial precondition for fostering participation is sharing control among the involved people (Fischer, 2007b). The roles that users can play vary, depending on their levels of involvement (Preece & Shneiderman, 2009). When users change their roles in the community by making constant contributions, they should be granted the matching authority in the decision-making process that shapes the system (Benkler & Nissenbaum, 2006). Responsibility without authority cannot sustain users' interest in further involvement. Giving people some authority is a further source of intrinsic motivation because it will attract and encourage new users who want to influence the system's development to make contributions.

Promote Mutual Learning and Support of Knowledge Exchange

Users have different levels of skill and knowledge about the system. To get involved in contributing to the system's evolution or using the system, they need to learn many things. Peer users are important learning resources. A meta-designed socio-technical environment should be accompanied by knowledge sharing mechanisms that encourage users to learn from each other. Therefore, a knowledge management infrastructure (as described previously) can be a STS by itself as well as a meta-tool to support the evolution of all kinds of STSs. For example, in open source software projects, mailing lists, discussion forums, and chat rooms provide important platforms for knowledge

transfer and exchange among peer users (Ye & Yamamoto, 2007).

Structure Communication to Support Reflection on Practice

Communication support has to be offered, which helps to combine usage of technical systems, collaboration, and design activities with mutual reflection. To fulfill Chern's (1987) principle that participants must reveal their assumptions, an appropriate communication structure is necessary. A facilitated communication that leaves enough time for reflection (e.g., by proceeding step-by-step), offers opportunities for the exchange of backgrounds and assumptions. Furthermore, within a culture of participation, users need to continuously see that their contributions make a recognizable influence on the system. Therefore, a communication procedure, such as the STWT (see the subsection "Structuring of Communication"), is feasible and makes the design artifacts (plans, models, etc.) continuously visible together with the improvements or proposals that are annotated by involved people. Considering an underdesigned plan of the socio-technical design step-by-step gives the participants sufficient time to reflect on it and to make their comments.

Complex design problems require more knowledge than any single person possesses. Therefore, knowledge exchange and construction among many domain experts must be fostered. Creating a shared understanding among domain experts requires facilitation so that different and often controversial points of view are brought together and lead to new insights, new ideas, and new artifacts.

CONCLUSION

New media and new technology provide new possibilities to rethink learning, working, and collaborating. In this article, we argue that new media and new technology on their own cannot support and transform these activities to meet the demands of the future, but that they have to be integrated into STSs.

Our analysis differentiates between a highest level of meta-design considerations, which cover a theoretical framework and its scientific substantiation, and an intermediate level that is represented by a meta-designed framework that includes concrete tools, procedures, methods, knowledge, and so forth. Within these frameworks, concrete socio-technical systems of a certain type can and do develop. They represent the basic level. The highest level—or meta level—is needed because it is not possible to provide a list of all concrete methods and tools that represent meta-design.

Socio-technical phenomena are self-referential: on the one hand, they are the outcomes of design and evolution, and on the other hand, they have the potential to support their own evolution. Collaboratories and knowledge management environments are typical examples. The strengths of socio-technical systems result from the integration of deterministic structures and processes and the contingency of social systems. Meta-design aims to support this integration.

Therefore meta-design offers a corridor by which the evolution and continuous adaptation, as is typical for social systems, can take place. Meta-design gives people who participate within a socio-technical system an opportunity to contribute to its evolutionary growth and to promote the evolution of their own social interactions. Therefore, the participant's work should be organized around seeds that represent boundary objects to which design can refer during use time. To avoid misunderstandings, we stress that the goal of meta-design is not to let untrained people develop and evolve sophisticated software systems, but to put *owners of problems* in charge. By contrast, the critical challenge is the creation of STSs that achieve the best fit between the technical components (mainly software and hardware) and their ever-changing context of use, problems, domains, users, and communities of users. Meta-design creates inherent tensions between *standardization* (which can suppress innovation and creativity) and *improvisation* (which can lead to a Babel of different and incompatible versions), and the success criteria

for meta-designed frameworks is whether they can balance this tension.

ACKNOWLEDGMENTS

REFERENCES

Ackerman, M. S. (1998). Augmenting organizational memory: A field study of answer garden. *ACM Transactions on Information Systems*, *16*(3), 203–224. doi:10.1145/290159.290160

Alavi, M., & Leidner, D. (2001). Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *Management Information Systems Quarterly*, *25*(1), 107–136. doi:10.2307/3250961

Appelt, W. (1999). WWW based collaboration with the BSCW system. In *SOFSEM '99: Proceedings of the 26th Conference on Current Trends in Theory and Practice of Informatics on Theory and Practice of Informatics* (pp. 66–78). London: Springer.

Arias, E., Eden, H., Fischer, G., Gorman, A., & Scharff, E. (2001). Transcending the individual human mind—creating shared understanding through collaborative design. In Carroll, J. M. (Ed.), *Human-computer interaction in the new millennium*. New York: ACM Press.

Avison, D. E., Lau, F., Myers, M. D., & Nielsen, P. A. (1999). Action research. *Communications of the ACM*, *42*(1), 94–97. doi:10.1145/291469.291479

Benkler, Y. (2006). *The wealth of networks: How social production transforms markets and freedom*. New Haven, CT: Yale University Press.

Benkler, Y., & Nissenbaum, H. (2006). Commons-based peer production and virtue. *Journal of Political Philosophy*, *14*(4), 394–419. doi:10.1111/j.1467-9760.2006.00235.x

Brand, S. (1995). *How buildings learn: What happens after they're built*. New York: Penguin Books.

Brown, J. S., Duguid, P., & Haviland, S. (1994). Toward informed participation: Six scenarios in search of democracy in the information age. *The Aspen Institute Quarterly*, *6*(4), 49–73.

Bryant, S. L., Forte, A., & Bruckman, A. (2005). Becoming Wikipedian: transformation of participation in a collaborative online encyclopedia. In *GROUP '05: Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work* (pp. 1–10). New York: ACM.

Carell, A., & Herrmann, T. (2009, June). Negotiation-tools in CSCL-scenarios—Do they have a valid use? In C. O'Malley, D. Suthers, P. Reimann, & A. Dimitracopoulou (Eds.), *Computer Supported Collaborative Learning Practices: CSCL2009 Conference Proceedings,* Rhodos, Greece (pp. 557–567). International Society of the Learning Sciences.

Carell, A., & Herrmann, T. (2010). Interaction and collaboration modes for integration inspiring information into technology-enhanced creativity workshops. In *Proceedings of the 43rd Hawaii International Conference on System Sciences*. Los Alamitos, CA: IEEE Computer Society Press.

Carmien, S., Dawe, M., Fischer, G., Gorman, A., Kintsch, A., Sullivan, J., & James, F. (2005). Socio-technical environments supporting people with cognitive disabilities using public transportation. *ACM Transactions on Computer-Human Interaction*, *12*(2), 233–262. doi:10.1145/1067860.1067865

Carmien, S. P., & Fischer, G. (2008). Design, adoption, and assessment of a socio-technical environment supporting independence for persons with cognitive disabilities. In *CHI '08: Proceedings of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems* (pp. 597–606). New York: ACM.

Carroll, J. M. (Ed.). (1995). *Scenario-based design for human computer interaction*. New York: John Wiley.

Checkland, P. (1981). *Systems thinking, systems practice*. Chichester, UK: John Wiley & Sons.

Cherns, A. (1976). The principles of sociotechnical design. *Human Relations*, *29*(8), 783–792. doi:10.1177/001872677602900806

Cherns, A. (1987). Principles of sociotechnical design revisted. *Human Relations*, *40*(3), 153–162. doi:10.1177/001872678704000303

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, *34*(11), 131–133. doi:10.1109/2.963450

Coiera, E. (2007). Putting the technical back into socio-technical systems research. *International Journal of Medical Informatics*, *76*, 98–103. doi:10.1016/j.ijmedinf.2006.05.026

dePaula, R. (2004). *The construction of usefulness: How users and context create meaning with a social networking system*. Unpublished doctoral dissertation, University of Colorado at Boulder.

dePaula, R., Fischer, G., & Ostwald, J. (2001). Courses as seeds: expectations and realities. In P. Dillenbourg, A. Eurelings, & K. Hakkarainen (Eds.), *Proceedings of the Second European Conference on Computer-Supported Collaborative Learning (Euro-CSCL'2001)* (pp. 494–501). Maastricht, The Netherlands: University of Maastricht.

Diefenbruch, M., Hoffmann, M., Misch, A., & Schneider, H. (2000, October). Situated knowledge management—On the borderline between chaos and rigidity. In U. Reimer (Ed.), *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management (PAKM 2000),* Basel, Switzerland (pp. 8-1–8-7). CEUR-WS.org.

Eason, K. (1988). *Information technology and organisational change*. London: Taylor & Francis.

Finholt, T., & Olson, G. (1997). From laboratories to collaboratories: A new organizational form for scientific collaboration. *Psychological Science*, *8*(1), 28–36. doi:10.1111/j.1467-9280.1997.tb00540.x

Fischer, G. (2002). Beyond "couch potatoes": From consumers to designers and active contributors. *First Monday*. Retrieved from http://131.193.153.231/www/issues/issue7_12/fischer/index.html

Fischer, G. (2003, June). Meta-design: Beyond user-centered and participatory design. In J. Jacko & C. Stephanidis (Eds.), *Proceedings of Human-Computer Interaction 2003 Conference,* Crete, Greece (pp. 88–92). Hillsdale, NJ: Lawrence Erlbaum.

Fischer, G. (2006). Distributed intelligence: Extending the power of the unaided, individual human mind. In *AVI '06: Proceedings of the Working Conference on Advanced Visual Interfaces* (pp. 7–14). New York: ACM.

Fischer, G. (2007a). Designing socio-technical environments in support of meta-design and social creativity. In *Proceedings of Conference on Computer Supported Collaborative Learning (CSCL'2007) New Brunswick, NJ* (pp. 1–10). International Society of the Learning Sciences.

Fischer, G. (2007b). Meta-design: expanding boundaries and redistributing control in design. In *INTERACT'07: Proceedings of the 11th IFIP TC 13 International Conference on Human-Computer Interaction,* Rio de Janeiro, Brazil (pp. 193–206). Berlin: Springer.

Fischer, G. (2010). End-user development and meta-design: Foundations for cultures of participation. *Journal of Organizational and End User Computing*, *22*(1), 52–82.

Fischer, G., & Giaccardi, E. (2006). Meta-design: A framework for the future of end-user development. In Lieberman, H., Paternò, F., & Wulf, V. (Eds.), *End user development* (pp. 427–457). Dordrecht, The Netherlands: Kluwer Academic Publishers. doi:10.1007/1-4020-5386-X_19

Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., & Sumner, T. (1998). Embedding critics in design environments. In Maybury, M. T., & Wahlster, W. (Eds.), *Readings in intelligent user interfaces* (pp. 537–561). San Francisco: Morgan Kaufmann Publishers.

Fischer, G., Nakakoji, K., & Ye, Y. (2009). Meta-design: Guidelines for supporting domain experts in software development. *IEEE Software*, *26*, 37–44. doi:10.1109/MS.2009.134

Fischer, G., & Ostwald, J. (2002, June). Seeding, evolutionary growth, and reseeding: Enriching participatory design with informed participation. In T. Binder, J. Gregory, & I. Wagner (Eds.), *Proceedings of the Participatory Design Conference (PDC'02),* Malmö, Sweden (pp. 135–143). CPSR.

Fischer, G., Piccinno, A., & Ye, Y. (2008). The ecology of participants in co-evolving socio-technical environments. In P. Forbrig & F. Paternò (Eds.), *Engineering Interactive Systems: Proceedings of the 2nd Conference on Human-Centered Software Engineering,* Pisa, Italy (LNCS 5247, pp. 279–286).

Forte, A., Larco, V., & Bruckman, A. (2009). Decentralization in Wikipedia governance. *Journal of Management Information Systems*, *26*(1), 49–72. doi:10.2753/MIS0742-1222260103

Fowler, M. (2001). The new methodology. *Wuhan University Journal of Natural Sciences*, *6*(1), 12–24. doi:10.1007/BF03160222

Giaccardi, E. (2004). *Principles of metadesign: Processes and levels of co-creation in the new design space.* Unpublished doctoral dissertation, CAiiA-STAR, School of Computing, Plymouth, UK.

Goedicke, M., & Herrmann, T. (2008). A case for viewpoints and documents. In Paech, B., & Martell, C. (Eds.), *Innovations for requirement analysis. From stakeholders' needs to formal designs* (pp. 62–84). Berlin: Springer. doi:10.1007/978-3-540-89778-1_8

Grudin, J., & Pruitt, J. (2002, June). Personas, participatory design and product development: An infrastructure for engagement. In T. Binder, J. Gregory, & I. Wagner (Eds.), *PDC 02: Proceedings of the Participatory Design Conference,* Malmö, Sweden (pp. 144–161).

Habermas, J. (1981). Theorie des kommunikativen Handelns.: *Vol. 1. Handlungsrationalität und gesellschaftliche Rationalisierung*. Berlin: Suhrkamp Verlag.

Habraken, J. (1972). *Supports: An alternative to mass housing. Tyne & Wear*. UK: Urban International Press.

Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, *8*, 231–274. doi:10.1016/0167-6423(87)90035-9

Henderson, A., & Kyng, M. (1991). There's no place like home: Continuing design in use. In Greenbaum, J., & Kyng, M. (Eds.), *Design at Work: Cooperative Design of Computer Systems* (pp. 219–240). Hillsdale, NJ: Lawrence Erlbaum Associates.

Herrmann, T. (2003, July). Learning and teaching in socio-technical environments. In T. J. V. Weert & R. K. Munro (Eds.), *Informatics and the digital society: Social, ethical and cognitive issues, Proceedings of SECIII 2002—Social, Ethical and Cognitive Issues of Informatics and ICT Conference,* Dortmund, Germany (pp. 59–72).

Herrmann, T. (2009). Systems design with the socio-technical walkthrough. In Whitworth, A., & de Moore, B. (Eds.), *Handbook of research on socio-technical design and social networking systems* (pp. 336–351). Hershey, PA: Idea Group Publishing.

Herrmann, T. (2010). Support of collaborative creativity for co-located meetings. In Randall, D., & Salembier, P. (Eds.), *From CSCW to Web 2.0*. Berlin: Springer.

Herrmann, T., & Hoffmann, M. (2005). The metamorphoses of workflow projects in their early stages. *Computer Supported Cooperative Work*, *14*(5), 399–432. doi:10.1007/s10606-005-9006-8

Herrmann, T., Hoffmann, M., Jahnke, I., Kienle, A., Kunau, G., Loser, K., et al. (2003a). Concepts for usable patterns of groupware applications. In M. Pendergast, K. Schmidt, C. Simone, & M. Tremaine (Eds.), *Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work* (pp. 349–358). New York: ACM Press.

Herrmann, T., Hoffmann, M., Kunau, G., & Loser, K. (2004). A modeling method for the development of groupware applications as socio-technical systems. *Behaviour & Information Technology, 23*(2), 119–135. doi:10.1080/01449290310001644840

Herrmann, T., Hoffmann, M., Loser, K., & Moysich, K. (2000). Semistructured models are surprisingly useful for user-centered design. In Dieng, R., Giboin, A., Karsenty, L., & De Michelis, G. (Eds.), *Designing cooperative systems: Proceedings of Coop 2000* (pp. 159–174). Amsterdam, The Netherlands: IOS Press.

Herrmann, T., Kienle, A., & Reiband, N. (2003). Meta-knowledge—a success factor for computer-supported organizational learning in companies. *Training Issues for Successful ICT Innovation in Companies of Educational Technology & Society, 6*(1), 9–13.

Herrmann, T., Kunau, G., Loser, K., & Menold, N. (2004, July). Sociotechnical walkthrough: Designing technology along work processes. In A. Clement, F. Cindio, A. Oostveen, D. Schuler, & P. van den Besselaar (Eds.), *Artful integration: Interweaving media, materials and practices. Proceedings of the eighth Participatory Design Conference 2004,* Toronto, ON, Canada (Vol. 1, pp. 132–141). New York: ACM Press.

Herrmann, T., & Loser, K. (1999). Vagueness in models of socio-technical systems. *Behavior & Information Technology: Special Issue on Analysis of Cooperation and Communication, 18*(5), 313–323.

Herrmann, T., Loser, K., & Jahnke, I. (2007). Socio-technical walkthrough (STWT): A means for knowledge integration. *International Journal of Knowledge and Organizational Learning Management, 14*(5), 450–464. doi:10.1108/09696470710762664

Kensing, F., & Blomberg, J. (1998). Participatory design: Issues and concerns. *Computer Supported Cooperative Work, 7*(3), 167–185. doi:10.1023/A:1008689307411

Kensing, F., Simonsen, J., & Bødker, K. (1996). MUST—A method for participatory design. In J. Blomberg, F. Kensing, & E. A. Dykstra-Erickson (Ed.), *Proceedings of the Participatory Design Conference (PDC '96)* (pp. 129–140).

Kirkeby, O. F. (2000). *Management philosophy*. Berlin: Springer.

Kunau, G. (2006). *Facilitating computer supported cooperative work with socio-technical self-descriptions*. Retrieved from http://hdl.handle.net/2003/22226

Latour, B. (1999). *Pandora's hope: Essays on the reality of science studies*. Cambridge, MA: Harvard University Press.

Lieberman, H., Paterno, F., & Wulf, V. (Eds.). (2006). *End user development*. Dordrecht, The Netherlands: Kluwer Publishers. doi:10.1007/1-4020-5386-X

Luhmann, N. (1995). *Social systems*. Stanford, CA: University Edition.

Maturana, H. R., & Varela, F. J. (1980). *Autopoiesis and cognition: The realization of the living*. Dordrecht, The Netherlands: Kluwer Publishers.

Mørch, A. (1997). Three levels of end-user tailoring: Customization, integration, and extension. In King, M., & Mathiassen, L. (Eds.), *Computers and Design in Context* (pp. 51–76). Cambridge, MA: MIT Press.

Mørch, A. I., Stevens, G., Won, M., Klann, M., Dittrich, Y., & Wulf, V. (2004). Component-based technologies for end-user development. *Communications of the ACM, 47*(9), 59–62. doi:10.1145/1015864.1015890

Mumford, E. (1987). Sociotechnical systems design. Evolving theory and practice. In Bjerknes, G., Ehn, P., & Kyng, M. (Eds.), *Computers and Democracy: A Scandinavian Challenge* (pp. 59–77). Aldershot, UK: Avebury.

Mumford, E. (2000). A Socio-technical approach to systems design. *Requirements Engineering, 5*, 125–133. doi:10.1007/PL00010345

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

O'Reilly, T. (2006). *What is Web 2.0—Design patterns and business models for the next generation of software*. Retrieved from http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

Orlikowski, W. J. (1992). The duality of technology: Rethinking the concept of technology in organizations. *Organization Science, 3*(3), 398–427. doi:10.1287/orsc.3.3.398

Orlikowski, W. J. (1996). Improvising organizational transformation over time: A situated change perspective. *Information Systems Research, 7*(1), 63–92. doi:10.1287/isre.7.1.63

Pedersen, P. P. (2000). Our present: Postmodern? In Andersen, H., & Kaspersen, L. B. (Eds.), *Classical and Modern Social Theory* (pp. 412–431). Oxford, UK: Blackwell Publishers.

Pinelle, D., & Gutwin, C. (2002). Groupware walkthrough: Adding context to groupware usability evaluation. *CHI Letters, 4*(1), 455–462.

Pipek, V. (2005). *From tailoring to appropriation support: Negotiating groupware usage.* Retrieved from http://herkules.oulu.fi/isbn9514276302/

Pipek, V., Rossen, M. B., deRuyter, B., & Wulf, V. (Eds.). (2009). *End-user development.* Berlin: Springer. doi:10.1007/978-3-642-00427-8

Polson, P. G., Lewis, C., Rieman, J., & Wharton, C. (1992). Cognitive walkthrough: a method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies, 36,* 741–773. doi:10.1016/0020-7373(92)90039-N

Preece, J., & Shneiderman, B. (2009). The reader-to-leader framework: Motivating technology-mediated social participation. *AIS Transactions on Human-Computer Interaction, 1*(1), 13–32.

Prilla, M. (2009). Models, social tagging and knowledge management? A fruitful combination for process improvement. In S. Rindele-Ma, S. Sadiq, & F. Leymann (Eds.), *Business Process Management Workshops. BPM 2009 International Workshops,* Ulm, Germany (LNBIP 43, pp. 266-277). Berlin: Springer.

Rittel, H. W., & Webber, M. M. (1973). Planning problems are wicked problems. In Cross, N. (Ed.), *Developments in design methodology* (pp. 135–144). Ann Arbor, MI: UMI Research Press.

Rogoff, B., Matusov, E., & White, C. (1998). Models of teaching and learning: Participation in a community of learners. In Olsen, D. R., & Torrance, N. (Eds.), *The handbook of education and human development: New models of learning, teaching and schooling* (pp. 388–414). Oxford, UK: Blackwell.

Scharff, E. (2002). *Open source software, a conceptual framework for collaborative artifact and knowledge construction.* Unpublished doctoral dissertation, University of Colorado at Boulder.

Schmidt, K. (1999). Of maps and scripts—The status of formal constructs in cooperative work. *Information and Software Technology, 41*(6), 319–329. doi:10.1016/S0950-5849(98)00065-2

Schön, D. A. (1983). *The reflective practitioner: How professionals think in action.* New York: Basic Books.

Simon, H. A. (1996). *The sciences of the artificial.* Cambridge, MA: MIT Press.

Star, S. L. (1989). The Structure of Ill-Structured Solutions: Boundary Objects and Heterogeneous Distributed Problem Solving. In *Distributed Artificial Intelligence, 2,* 37–55. San Francisco: Morgan Kaufmann Publishers

Suchman, L. A. (1987). *Plans and situated actions.* Cambridge, UK: Cambridge University Press.

Tapscott, D., & Williams, A. D. (2006). *Wikinomics: How mass collaboration changes everything. portofolio.* New York: Penguin Group.

Thoresen, K. (1997). Workflow meets work practice. *Accounting Management and Information Technologies, 7*(1), 21–36. doi:10.1016/S0959-8022(97)00002-7

Trist, E. (1981). The evolution of socio-technical systems. In Van de Ven, A. H., & Joyce, W. F. (Eds.), *Perspectives on organization design and behavior.* New York: Wiley.

Von Bertalanffy, L. (1973). *General system theory: Foundations, development, applications.* New York: G. Braziller.

Whitworth, B. (2009). The social requirements of technical systems. In Whitworth, B., & de Moor, A. (Eds.), *Handbook of Research on Socio-Technical Design and Social Networking Systems* (pp. 3–22). Hershey, PA: IGI Global.

Winograd, T., & Flores, F. (1986). *Understanding computers and cognition: A new foundation for design.* Norwood, NJ: Ablex Publishing Corporation.

Wright, M., Marlino, M., & Sumner, T. (2002). Meta-design of a community digital library. *D-Lib Magazine, 8*(5). doi:10.1045/may2002-wright

Wulf, V., & Rohde, M. (1995). Towards an integrated organization and technology development. In *Proceedings of the Symposium on Designing Interactive Systems,* Ann Arbor, MI (pp. 55–64). New York: ACM-Press.

Ye, Y., Yamamoto, Y., & Nakakoji, K. (2007). A socio-technical framework for supporting programmers. In *Proceedings of the 2007 ACM Symposium on Foundations of Software Engineering (FSE2007),* Dubrovnik, Croatia (pp. 351–360). New York: ACM Press.

Yourdon, E. (1979). *Structured walkthroughs.* Upper Saddle River, NJ: Prentice Hall.

Yu, E. S. K., & Mylopoulos, J. (1994). Understanding "why" in software process modeling, analysis, and design. In *Proceedings of the 16th International Conference on Software Engineering,* Sorrento, Italy (pp. 159–168). Los Alamitos, CA: IEEE Computer Society Press.