Position Paper for the **Design Requirements Workshop**, June 3 – 6, 2007, Case Western Reserve University, Cleveland, Ohio, USA

Meta-Design: Coping with Ill-Defined Problems and Emerging Requirements in Collaborative Design

Gerhard Fischer
University of Colorado, Center for LifeLong Learning and Design (L3D)
Department of Computer Science, 430 UCB
Boulder, CO 80309-0430 – USA
gerhard@colorado.edu

1. Design Problems

Design Problems are Unique

Design problems (in contrast to problems in the natural science) are *unique*: they form a "universe of one". Design projects require learning and produces new knowledge in the form of understanding as well as artifacts. Complexity in design arises from the need to synthesize stakeholders' different perspectives of a problem, the management of large amounts of information relevant to a design task, and understanding the design decisions that have determined the long-term evolution of a designed artifact.The uniqueness requires different methodologies to approach and evaluate design problems, including different choices on how to find the most appropriate mixture between relevance and rigor [Schön, 1983].

Design Problems require the Integration of Problem Framing and Problem Solving

Design problems are *"wicked"* in the sense of defying logi"al approaches and optimal solutions [Rittel & Webber, 1984]. The idea of finding an optimal solution to design problems was abandoned by Simon in his introduction of the concept of *"satisficing"* [Simon, 1996] as an alternative to optimizing. A satisficing approach attempts to find the best possible compromise between design constraints when an optimal solution is impossible.

Wicked problems are not analyzed in one step and then enacted in the next. Instead, the process of problem framing and problem solving needs to be intertwined. The role of the designer needs to be expanded from being solely a problem solver, to also being a "problem framers" playing an active role in defining the problem to be solved. The dualism of design states that one cannot understand a problem without having a concept of the solution in mind: *"one cannot gather information meaningfully unless one has understood the problem but one cannot understand the problem without information about it"* [Rittel & Webber, 1984].

Emphasizing the integration of problem framing and problem solving casts design as a search *for* a problem space rather than just *within* a problem space. It brings into question all design methodologies that are founded on a separation of analysis and synthesis. Furthermore, it emphasizes the importance of problem owners (those for whom an artifact is designed) as stakeholders in the design process because they have the authority and the knowledge to reframe the problem as the problem space becomes better understood.

Design Problems transcend the Individual Human Mind

Most previous design research has focused on support for individual reflective practitioners. The world, however, has become too complex for individuals to have the knowledge necessary to tackle problems by themselves. The knowledge relevant to complex problems is *distributed among many people*. This requires socio-technical environments that can bring together people from different domains and with diverse perspectives and can enable and facilitate *reflective communities* that collectively reflect, learn, and act upon shared design situations.

Our research interest is in designing the social and technical infrastructures in which *new forms of collaborative design* can take place. For most of the design domains (ranging from urban design to graphics and software design) that we have studied over many years [Arias et al., 2000], the knowledge to understand, frame, and solve problems is not given, but is constructed and evolves during the problem-solving process.

Design Communities

Design communities are social structures that enable groups of people sharing knowledge and resources in support of collaborative design. Different communities grow around different types of design practice. Each design community is unique, but for the purposes of this discussion, we identify two stereotypical kinds of design community—communities of practice (CoPs) and communities of interest (CoIs)—and discuss their respective barriers and biases for knowledge creation and sharing in collaborative design [Fischer, 2001].

Communities of Practice (CoPs). CoPs [Wenger, 1998] consist of practitioners who work as a community in a certain domain undertaking similar work. Learning within a CoP takes the form of *legitimate peripheral participation* (LPP), which is a type of apprenticeship model in which newcomers enter the community from the periphery and move toward the center as they become more and more knowledgeable,

Sustained engagement and collaboration lead to boundaries that are based on shared histories of learning and that create discontinuities between participants and non-participants. Highly developed knowledge systems (including conceptual frameworks, technical systems, and human organizations) are biased toward efficient communication *within* the community at the expense of acting as barriers to communication with outsiders: boundaries that are empowering to the insider are often barriers to outsiders and newcomers to the group.

A community of practice has many possible paths and many roles (identities) within it (e.g., leader, scribe, power-user, visionary, and so forth). Over time, most members move toward the center, and their knowledge becomes part of the foundation of the community's shared background.

Communities of Interest (CoIs). CoIs bring together stakeholders from different CoPs and are defined by their collective concern with the resolution of a particular problem. CoIs can be thought of as "communities of communities" or a community of representatives of communities. Examples of CoIs are: (1) a team interested in software development that includes software designers, users, marketing specialists, psychologists, and programmers, or (2) a group of citizens and experts interested in urban planning, especially with regard to implementing new transportation systems.

Stakeholders within CoIs are considered as *informed participants* who are neither experts nor novices, but rather both: they are experts when they communicate their knowledge to others, and they are novices when they learn from others who are experts in areas outside their own knowledge.

The fundamental barrier facing CoIs is that knowledge distribution is based on a *symmetry of ignorance* [Fischer, 2000] in which each stakeholder possesses some, but not all, relevant knowledge, and the knowledge of one participant complements the ignorance of another. This barrier must be overcome by building a shared understanding of the task at hand, which often does not exist at the beginning, but evolves incrementally and collaboratively and emerges in people's minds and in external artifacts. Members of CoIs must learn to communicate with and learn from others who have different perspectives and perhaps a different vocabulary for describing their ideas. In other words, this symmetry of ignorance must be exploited.

2. Meta-Design

In a world that is not predictable, improvisation, evolution, and innovation are more than a luxury: they are a necessity. The challenge of design is not a matter of getting rid of the emergent, but rather of including it and making it an opportunity for more creative and more adequate solutions to problems.

Meta-design [Fischer & Giaccardi, 2006; Fischer et al., 2004] is an emerging conceptual framework aimed at defining and creating social and technical infrastructures in which new forms of collaborative design can take place. It extends the traditional notion of system design beyond the original development of a system. It is grounded in the basic assumption that future uses and problems cannot be completely anticipated at design time, when a system is developed. Users, at use time, will discover mismatches between their needs and the support that an existing system can provide for them. These mismatches will lead to breakdowns that serve as potential sources of new insights, new knowledge, and new understandings.

Consumers and Designers

Cultures are substantially defined by their media and their tools for thinking, working, learning, and collaborating. A great amount of media is designed to see humans only as consumers. The importance of meta-design rests on the fundamental belief that humans (not all of them, not at all times, not in all contexts) want to be and act as designers in personally meaningful activities. Meta-design encourages users to be actively engaged in generating creative extensions to the artifacts given to them and has the potential to break down the strict counterproductive barriers between consumers and designers.

Many computer users and designers today are domain professionals, competent practitioners, and discretionary users, and should not be considered as naive users or "dummies." They worry about tasks, they are motivated to contribute and to create good products, they care about personal growth, and they want to have convivial tools that make them independent of "*high-tech scribes*" (whose role is defined by the fact that the world of computing is still too much separated into a population of elite scribes who can act as designers and a much larger population of intellectually disenfranchised computer phobes who are *forced* into consumer roles). The experience of having participated in the framing and solving of a problem or in the creation of an artifact makes a difference to those who are affected by the solution and therefore consider it personally meaningful and important.

A fundamental challenge for the next generation of computational media and new technologies is not to deliver predigested information to individuals, but to provide the opportunity and resources for social debate, discussion, and collaborative design. In many design activities, learning cannot be restricted to finding knowledge that is "out there." For most design problems (ranging from urban design to graphics design and software design, which we have studied over many years), the knowledge to understand, frame, and solve problems does not exist; rather, it is constructed and evolved during the process of solving these problems, exploiting the power of *"breakdowns"*. From this perspective, *access* to existing information and knowledge (often seen as the major advance of new media) is a very limiting concept.

Unself-conscious and Self-conscious Design Cultures

The theory of *self-conscious* and *unself-conscious* design cultures [Alexander, 1964] provides an initial analytical framework for gaining a systematic understanding of the fundamental difference between domain experts and software professionals.

Self-conscious design culture. Dictated by a self-conscious design culture, the major focuses of software engineering research are understanding, representing correctly, and satisfying what the users want; creating software systems that have high production values; and providing the development process that achieves the highest economic efficiency and that is repeatable. The distinct separation of users and developers is one of the most important tacit assumptions underlying software engineering research.

Unself-conscious design cultures. Domain experts who engage in software development activities are not interested in the system per se, but rather in the domain-specific tasks that have to be performed with the help of the system. For them, because they are not professional software developers, software systems are tools, and the introduction of new tools changes the tasks and practices, which in turn begets new needs for tools. This *co-evolution* of tools and tasks determines that a large class of software systems can never be completely delegated to external professional software developers, and can be developed only by those domain experts who own the problem and have both the inside knowledge of the application domain and software development skills.

Understanding the Impact of Meta-Design on Requirements

Requirements research should explore the following hypotheses/claims/opportunities:

- **Hypothesis$_1$: Requirements are generated differently.** Because developers are users themselves, there is not always a need for an elaborate requirement analysis phase as a major activity preceding the construction of the software system. Rapid changes of requirements need not be avoided; quite to the contrary, they are desired because the computer in such contexts is used to explore new possibilities and to find the "undreamed-of requirements"
- **Hypothesis$_2$: Software testing is conducted differently**. Because domain expert developers themselves are the primary users, complete testing is not as important as in the case when the developers are not the users.
- **Hypothesis$_3$: Collaboration takes place along different dimensions**. In self-conscious software development, a team of developers is often organized before the project starts — in unself-conscious software development, a predefined project team does not exist. Collaboration is spontaneous and opportunistic rather than planned.
- **Hypothesis$_4$: The path to the acquisition of knowledge and skill for software development is different**. Due to the lack of interest in software per se and the lack of professional training, domain experts are more likely to acquire software knowledge in a piecemeal fashion and demand-driven manner. Their knowledge is more fragmental than systematic.
- **Hypothesis$_5$: Software will evolve in a different style**. The system is evolved gradually by a large number of people who make small contributions each time. Evolution is more spontaneous and situational due to the co-adaptivity of tools and their users.

The Seeding, Evolutionary Growth, Reseeding (SER) Model

The seeding, evolutionary growth, reseeding (SER) model [Fischer & Ostwald, 2002] addresses challenging unresolved issues by demonstrating that there are no real borders between design practice and practice of use, but that these phases are highly related if meta-design is supported. The SER model postulates that systems that evolve over a sustained time span must continually alternate between periods of activity and unplanned evolution, and periods of deliberate (re)structuring and enhancement. We have explored the feasibility and usefulness of the SER model in the development of domain-oriented design environments, organizational memories, course information environments, and open systems approaches. The evolutions of these systems share common elements, all of which relate to sustained knowledge use and construction in support of informed participation.

Seeding. In the past, large and complex information systems were built as "complete" artifacts through large efforts of a small number of people. Conversely, instead of attempting to build complete systems, the SER model advocates building seeds that can be evolved over time through small contributions of a large number of people.

A *seed* is an initial collection of domain knowledge that is designed to evolve. It is created by environment developers and future users to be as complete as possible. However, no collection of knowledge can be truly complete due to the situated and tacit nature of knowledge as well as the constant change occurring in the environment in which the system is embedded [Suchman, 1987; Winograd & Flores, 1986]. No absolute requirements exist for the completeness, correctness, or specificity of the information in the seed. In fact, it is often shortcomings in these respects that provoke knowledge workers to add new information to the seed.

Evolutionary Growth. The evolutionary growth phase is one of decentralized evolution as knowledge workers use and extend the seed to do work or explore a problem [Henderson & Kyng, 1991]. During this phase, the seed plays two roles simultaneously: (1) it provides resources for work (information that has been accumulated from prior use), and (2) it accumulates the products of work, as each project contributes new information to the seed. During the evolutionary growth phase, workers are focused on solving a specific problem, rather than on creating reusable information. As a result, the information added during this phase will not be well integrated with the rest of the information in the seed.

Reseeding. Reseeding is a deliberate and centralized effort to organize, formalize, and generalize knowledge created during the evolutionary growth phase [Shipman & McCall, 1994]. The goal of reseeding is to create an information space in which useful information can be found, reused, and built upon. As in the seeding phase, users must participate in reseeding, since they can best judge what information is useful and what structures will serve their work practice.

Reseeding is necessary when evolutionary growth no longer proceeds smoothly. It is an opportunity to assess the information created in the context of specific projects and activities, and to decide what should be incorporated into a new seed to support the next cycle of evolutionary growth and reseeding. For example, open source software systems [Raymond & Young, 2001] often evolve for some time by adding patches, but eventually a new major version that incorporates the patches in a coherent fashion must be released.

3. Elements of *a 10-year Agenda for Research in Design Requirements Practices*

Trade-off between Standardization and Improvisation

Meta-design creates an inherent tension between standardization and improvisation. The SAP Info (from July 2003, page 33) argues to reduce the number of customer modifications for the following reasons: *"every customer modification implies costs because it has to be maintained by the customer. Each time a support package is imported there is a risk that the customer modification may have to be adjusted or re-implemented. To reduce the costs of such on-going maintenance of customer-specific changes, one of the key targets during an upgrade should be to return to the SAP standard wherever this is possible".* Finding the right balance between standardization (which can suppress innovation and creativity) and improvisation (which can lead to a Babel of different and incompatible versions) has been noted as a challenge in open source environments in which forking has often led developers in different directions.

From "Ease-of-Use" to "Low Threshold and High Ceiling"

"Ease-of-use" along with the "burden of learning something" are often used as arguments for why people will not engage in design. Building systems that support users to act as designers and not just as consumers is often less successful than the meta-designers have hoped for.

The end-user modifiability and end-user programming features themselves add often considerably more functionality to already very complex environments (such as high functionality applications and large software reuse libraries) — and our empirical analyses clearly show that not too many users of such systems are willing to engage in this additional learning effort.

Based on our work with user communities, it is obvious that serious working and learning do not have to be unpleasant — they can be empowering, engaging, and fun. Many times the problem is not that *programming is difficult, but that it is boring* (as we were told by an artist). Highly creative owners of problems struggle to learn tools that are useful to them, rather than believing in the alternative of "ease-of-use," which limits them to preprogrammed features.

Motivation and Rewards

What makes people, over time, become active contributors and designers and share their knowledge requires a new "design culture", involving a mindset change and principles of social capital accumulation. But before new social mindsets and expectations emerge, users' active participation comes as a function of simple motivational mechanisms and activities considered personally meaningful.

One focus of meta-design is the design of socio-technical environments in which interactive systems are embedded, and in which users are recognized and rewarded for their contributions and can accumulate social capital. *Social capital* is based on specific benefits that flow from the trust, reciprocity, information, and cooperation associated with social networks.

From Reflective Practitioners to Reflective Communities

To effectively support reflective communities—not just individual reflective practitioners [Schön, 1983]— the distances existing among its members must be overcome and exploited as opportunities. To effectively support the exchange of information needed for collaborative problem framing and solving and increase the participation of all stakeholders [Brown et al., 1994] in the design problems that affect them, we must eliminate some of the barriers produced by *spatial* distances ("voices from far away"), *temporal* distances ("voices from the past"), *conceptual* distances ("voices from people with different knowledge and backgrounds"), and *technological* distances ("virtual voices embedded in the artifacts") [Fischer, 2005]. This is a fundamental challenge for the design of human-centered computing of the future.

To support *design by a reflective community*, our research is focused on:

- creating a *common design workspace* that ensures the continuity of the cycles of action and reflection in a reflective community in which multiple designers can engage in actions in a seamless yet coherent way;
- ensuring the *visibility* of one member's action so as to bring an empathic experiential mode in other designers and trigger the reflection of other members *in* action as well as *on* action (reflection is both experiential and rational as it takes place both in action and on action); and
- *intertwining individual reflection and collective reflection* so that the reflection of each member can be channeled back into the common design workspace to influence subsequent design actions.

When a reflective community engages in collaborative design through the cycles of collective action and reflection, we need to pay attention to the multidimensional distances that hinder or strengthen the synthesis of action and reflection from multiple agents and resources. In particular, we need to gain a better understanding of the barriers and opportunities represented by conceptual and technological dimensions of distribution.

4. References

Alexander, C. (1964) *The Synthesis of Form,* Harvard University Press, Cambridge, MA.

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

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), pp. 49-73.

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

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

Fischer, G. (2005) "Distances and Diversity: Sources for Social Creativity," *Proceedings of Creativity & Cognition*, London, April, pp. 128-136.

Fischer, G., & Giaccardi, E. (2006) "Meta-Design: A Framework for the Future of End User Development." In H. Lieberman, F. Paternò, & V. Wulf (Eds.), *End User Development: Empowering People to Flexibly Employ Advanced Information and Communication Technology,* Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 427-457.

Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A. G., & Mehandjiev, N. (2004) "Meta-Design: A Manifesto for End-User Development," *Communications of the ACM*, 47(9), pp. 33-37.

Fischer, G., & Ostwald, J. (2002) "Seeding, Evolutionary Growth, and Reseeding: Enriching Participatory Design with Informed Participation," *Proceedings of the Participatory Design Conference (PDC'02),* Malmö University, Sweden, pp. 135-143.

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

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

Rittel, H., & Webber, M. M. (1984) "Planning Problems are Wicked Problems." In N. Cross (Ed.), *Developments in Design Methodology,* John Wiley & Sons, New York, pp. 135-144.

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

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

Simon, H. A. (1996) *The Sciences of the Artificial,* third ed., The MIT Press, Cambridge, MA.

Suchman, L. A. (1987) *Plans and Situated Actions,* Cambridge University Press, Cambridge, UK.

Wenger, E. (1998) *Communities of Practice — Learning, Meaning, and Identity,* Cambridge University Press, Cambridge, UK.

Winograd, T., & Flores, F. (1986) *Understanding Computers and Cognition: A New Foundation for Design,* Ablex Publishing Corporation, Norwood, NJ.