

Meta-Design: A Conceptual Framework for End-User Software Engineering

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. Summary of own most relevant work for EUSE

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 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 understanding.

Consumers and Designers

Cultures are substantially defined by their media and their tools for thinking, working, learning, and collaborating. A great amount of new 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 naïve 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 *unself-conscious* and *self-conscious* design cultures (C. Alexander) 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 and many research problems framed under this assumption.

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.

2. Future Questions for EUSE

Understanding the Impacts of Meta-Design on Software Development

EUSE research should explore the following hypotheses / claims:

- **Hypothesis₁: Requirements are generated differently.** Because developers are users themselves, there is no 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 the new possibilities and to find the “undreamed-of requirements”
- **Hypothesis₂: 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.
- **Hypothesis₃: 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.
- **Hypothesis₄: 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.
- **Hypothesis₅: 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.

Trade-off between Standardization and Improvisation

Meta-design creates an inherent tension between standardization and improvisation. The SAP Info (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 and 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

Additional Topics (only enumerated here)

1. relationship between: end-user development, end-user software engineering, meta-design, web 2.0 approaches;
2. the relevance of EUSE as a contribution to a “science of design”;
3. support for the “seeding/location/comprehension/modification/sharing (sLCMS)” model;
4. putting owners of problems in charge by redefining the roles of high-tech scribes;
5. relationship between EUSE and different design methodologies (e.g., professionally dominated design, user-centered design, participatory design, learner-centered design);
6. EUSE does not only require reflective practitioners but *reflective communities*.

3. References Relevant to EUSE

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.

<http://l3d.cs.colorado.edu/~gerhard/papers/EUD-meta-design-online.pdf>

Fischer, G., Giaccardi, E., Eden, H., Sugimoto, M., & Ye, Y. (2005) “Beyond Binary Choices: Integrating Individual and Social Creativity,” *International Journal of Human-Computer Studies (IJHCS) Special Issue on Computer Support for Creativity* (E.A. Edmonds & L. Candy, Eds.), 63(4-5), pp. 482-512.

<http://l3d.cs.colorado.edu/~gerhard/papers/ind-social-creativity-05.pdf>

Fischer, G. (2005) "From Reflective Practitioners to Reflective Communities." In: Proceedings of the HCI International Conference (HCII), Las Vegas, July 2005, (published on CD).
<http://l3d.cs.colorado.edu/~gerhard/papers/reflective-communities-hcii-2005.pdf>

Fischer, G. (2005) "Computational Literacy and Fluency: Being Independent of High-Tech Scribes." In J. Engel, R. Vogel, & S. Wessolowski (Eds.), *Strukturieren - Modellieren - Kommunizieren. Leitbild mathematischer und informatischer Aktivitäten*, Franzbecker, Hildesheim, pp 217-230;
<http://l3d.cs.colorado.edu/~gerhard/papers/hightechscribes-05.pdf>

Ye, Y., & Fischer, G. (2005) "Reuse-Conducive Development Environments," *International Journal Automated Software Engineering*, Kluwer Academic Publishers, Dordrecht, Netherlands, 12(2), pp. 199-235
<http://l3d.cs.colorado.edu/~gerhard/papers/J-ASE-final.pdf>

Fischer, G. (2002): "Beyond 'Couch Potatoes': From Consumers to Designers and Active Contributors", in FirstMonday (Peer-Reviewed Journal on the Internet),
http://firstmonday.org/issues/issue7_12/fischer/