Scandinavian Journal of Information Systems

Volume 10 | Issue 1 Article 14

1998

Work. Computers. Design.

Frieder Nake
University of Bremen, nake@informatik.uni-bremen.de

Work. Computers. Design.

Frieder Nake
University of Bremen, Germany
nake@informatik.uni-bremen.de

Three words constitute the title of this short contribution: work, computers, and design. Not very exciting. The words are put in a sequence to indicate some transition – a transition that is strongly noticeable in the work of Morten Kyng. This note is in his honor. It should be read as a side remark to the publication of his thesis summary in this issue of the *Journal*. A few trivial remarks will introduce the three title words.

1.

The ultimate economic reason for software development is the mechanization of mental labor and intellectual *work* (Nake 1992). No matter what the peculiar facet of software artifacts may be that we study, there is no escape from that most general purpose of mechanization – even if, and though, we do not observe it on the surface of systems. To the extent, however, that software takes on the form of digital media, the concept of work alone does not cover the *raison d'être* of software well enough. "Activity" appears to be a more appropriate term, and "play" emerges as a category besides "work". In fact, "play" appears as a dimension of work requesting more careful attention. Ehn was one of the first to point that out (Ehn 1988). Keller did an empirical study on it (Keller 1998).

All work processes, in order to be performed efficiently, depend on tools and machines. Machines, on the other hand, get constructed and built only through work. The application of machinery in turn degrades, dequalifies, and replaces labor (Braverman 1974).

© Scandinavian Journal of Information Systems, 1998, 10(1&2):53–60

2.

Computers are a special kind of machines. They are needed to apply software to a situation of practice. Computers are machines to compute computable functions with. The development of software for everyday use and the broad dissemination of such software created a phenomenon of wide-ranging cultural dimension: computers became a commodity. As such, they completely turned over the technological infrastructure of society, a process that had started long before.

Two important aspects accompany this development: the appearance of the computer as *medium* (Andersen et al. 1993), and *interaction* as the prevailing mode of using the computer. More precisely, I view the computer as an instrumental medium (Schelhowe 1997), i.e. a medium that is turned into an instrument without any important outside change.

3.

These two aspects bring to the foreground the eminent relevance of *design* in the broadest sense. Software development has become a design problem of a new dimension – engineering design and aesthetic design merging. It appears trivial to consider software development as an *engineering* design task. It is not as trivial to consider it as a task of *aesthetic* design at the same time. But this double view has become necessary for the development of *good* (successful, convincing, useful and useable) systems. This double view obviously calls for dedicated, sincere interdisciplinary collaboration, even more: it calls for *trans*disciplinary work. Such cannot be achieved by individual acts of free will. It will be the result of lasting efforts by many.

The interactive use of computers is a qualitatively new mode of people encountering the computer. In a system comprised of one or several humans interacting via software with a computer, the confines inherent to computability may be levered (Wegner 1997). In order to design such systems, a new understanding of the difficult relation of formal computer artifacts (tools) and informal human activities is in need. In labor's fight against automation, an important request was that not everything that *could* be automated *should* be automated. Today, the rationale for such a guiding principle has shifted from political and economical reasons to design.

Context, situation, and design have become the catchwords of an important and influential part of research (viz. (Suchman 1987, Greenbaum & Kyng 1991, Nardi 1996, Winograd 1996)). Design is what emerges beyond the critique of information technology: design in postmodern times (Coyne 1995). The Scandinavian tradition of participation, work-orientation, and social responsibility has paved the way for this, perhaps without being fully aware of the implications. Morten Kyng is one of the outstanding proponents of that tradition.

4.

By the mid-eighties, it had become convincingly clear that design of computer systems was a task that strongly challenged traditional engineering approaches to the construction of technical systems. "Traditional approach" meant first,

to identify and separate a number of parameters governing (as one must believe in this approach) a real world situation; second, to turn these parameters into formal and quantitative variables; and third, to optimize some sort of behavior of the system by tuning the variables.

On a philosophical level this meant that the rationalistic assumption of separating the object from the subject, an assumption underlying the traditional approach, was challenged by the practical task of designing computer artifacts. Design now appeared as a complex cooperative activity that changed conditions and requirements of the problem setting *during* the process of solving the problem. The identity of the problem turned out to be a fiction. What a horrible situation for Cartesian thought! Characterizing this situation as one of "moving targets" almost sounds like a euphemism.

5.

A wealth of books and papers has appeared in the mid-eighties making clear that the basic rationalistic assumption did not work on problems of design for complex real world processes. Design of any but the most trivial system was inherently ill-defined. An entirely new approach was needed if the design of computer software should lead to better quality of work and product.

In the US, books by Winograd & Flores (1986), and by Dreyfus & Dreyfus (1986), later those by Suchman (1987), Laurel (1990), Norman (1993), and others, became influential. In Europe, the Aarhus decennial conferences, and the 1988 meeting in Germany on *Software Development and Reality Construction*, defined landmarks. They all combined a critique of assumptions underlying traditional software design with first proposals of a new approach. A new, and perhaps surprising reception of Heidegger's and Wittgenstein's philosophies, besides great attention for activity theory and the constructivist attitude towards reality, played an outstanding role in laying ground to this anti-rationalistic critique.

6.

Those books began to appear at a time when in Scandinavia a trade union based tradition had already reached its climax. Influenced in part by, but different from, attempts in England and continental Europe to involve users of computing equipment in the process of designing such artifacts, researchers in Scandinavia had established participatory design, and the collective resource approach, as a *soft science*. It promised to open a totally new avenue to the design of complex systems. This type of science is marked not by definitions, theorems, and proofs, but rather by experiment, exploration, cooperation and conflict. Systems were not only technically and socially determined, as if these were two separable aspects, they were intrinsically identified by the dialectics of technical means, individual skills, group behavior, and social context.

It became decisive to firmly root design of artifacts within the context of a social situation, rather than construct a system from a context-independent blueprint. Doing this on a serious scientific level enticed to accept a softening of the notion of "science". It should not come as a surprise that, as a consequence, a

sort of battle between "hard" and "soft" science was started.

7.

The current situation in system design is characterized by a tremendous amount of work, both practical and theoretical, going on in North America, Europe, and Japan which comes under the headings of Computer Supported Cooperative Work (CSCW), Human Computer Interaction (HCI), Information Systems, Participatory Design, or Multi- and Hypermedia. In one way or another, much of that research aims at a new theory of design, and an understanding of complex processes involving humans and artifacts.

As already indicated above, the systemic approach takes both, human actor and artificial creation, as elements of the system. Such an approach would have been intolerable to the left in the seventies, for humans were not elements of systems, they were living creatures with an interest to survive and evolve. Now, it seems, under the principle (should we say "paradigm"?) of design the systemic view loses some of its inhumane implications.

8.

Contributions to the field are numerous, and deviate from classical engineering work: much of the work consists of describing scenarios and developing prototypes. These allow possible future actors (called "users") to gather some kind of pre-experience of an artifact yet to be constructed. Such surrogate experience is discussed and evaluated by both the actors and designers. This, in turn, leads to the development of improved prototypes, and often to new sets of criteria for their evaluation.

9.

The dilemma typical for the design situation is that design always projects into the future, but can achieve well-tested results only by changing the present. As long as the artifact is a well isolated piece to be introduced into a well known situation detached reflection and construction seem to work. As soon as the artifact takes on a processual and semiotic nature (Nake 1994), and as it makes sense only as part of the situation it is about to change, reflection-in-action (Donald Schön's concept) becomes necessary, and design becomes more than pure technical construction. Since computer artifacts, beyond their functional attributes, are largely characterized by their use attributes, the design dilemma became an issue of great importance in system development research.

Design of computer artifacts is largely design of processes, i.e. of intangible products. What gets designed are descriptions. Design takes place within the semiotic dimension. The artifact to be designed as software has taken on the form of a medium, the medium has taken on the form of a process. The transformation of things into signs opens up for design the dimension of interpretation: what has to be designed is not a process of only using materials or tools, but a process of interpreting signs. This appears to be the essence of design in postmodern times.

10.

Morten Kyng looks at "design in context" and at "contexts for design". With this distinction, he puts emphasis on the dialectics of design that are often neglected: design is, on one side, a process involving organizational and societal interests that extend beyond, but also define, the artifact. Design is, on the other side, a technical practice carried out within a particular work setting. "Design in context", according to Kyng, requires new techniques for the cooperative engagement of end users, while at the same time that engagement requires new "contexts for design".

In accordance with the Scandinavian tradition and with the new understanding of design, Kyng's contribution is practical and theoretical at the same time – where "practical" should be read as "establishment of prototypes", and "theoretical" should be read as "non-formal definition of principles".

11.

During its short history of just about fifty years, computer science has undergone dramatic changes. In its early phase entirely oriented towards the machine, computer science now encompasses software design in the US, and has become informatics in most of Europe and other continents. The change in name is indicative of a radically different approach to the type of complex systems that are the subject matter of software design. Software design is no longer solely a matter of engineering expertise. It has rather developed into an intricate cooperative effort that has to take into account formal models, psychological aspects, aesthetic features, workplace considerations, work organization, and social and political implications.

Informatics has turned out to be a science defying the distinction of hard and soft science: informatics is both at the same time.

Roughly speaking, this insight is the result of the 25-years period from 1970 to 1995. This period is marked by the world-wide establishment and growth of informatics departments. Morten Kyng has actively contributed to the discourse of informatics during the time span from 1980 to 1995, playing a central role in developing the field within Scandinavia and Europe, and in establishing in the small, but increasingly influential, participatory design community in the United States.

12.

Kyng's contribution is on a practical as well as a reflective level. From the traditional design disciplines, as well as from anthropology and psychology, Morten Kyng has taken up the eminent role that context and situation play in the design and application of any artifact. Design is always a contextual problem: design is central to software development, and context is crucial to design. Morten Kyng has beautifully and convincingly demonstrated what such a general orientation could mean in practice. He and his colleagues have developed, and put into practical use, a number of tools and techniques that should be helpful for the design of computer artifacts of double quality: quality of product and process.

Kyng's long lasting contributions gain in strength by becoming more focussed. During their early years, they found a particularly friendly and fruitful environment for collaboration of workers, researchers, and trade unions. In their early successes, they proved very generally the point that complex computer artifacts cannot be designed in isolation. The users had to be involved. But users were experts themselves. Therefore participation turned into cooperation. The content of such cooperation is much more than getting together in an attempt to mutually agree on some decision. It is to unleash the specific skills and knowledge that those involved bring into this conflictuous effort. To allow for conflict is one of the lessons that Kyng has taught us.

The early invention of mock-ups and the application of prototypes as means of making our ideas and intentions clear have turned out to be powerful techniques in design. During the course of Kyng's research, the design process itself has become the object of study. This is demonstrated very clearly in the most recent publications.

13.

Kyng's contribution to the science of software thus is that context is not an abstract notion but a contradictory process of real life; users are not models, or feature vectors, but living beings with their interests; design is not a detached activity performed by some distant expert; and representation of ideas and of possible new realities is crucial to a successful process of design that enhances qualities of work and product.

All these insights now belong to the science of informatics. Morten Kyng is one of those who have persistently pursued this line of thought and have advanced it to international acceptance and reputation. During this pursuit, computer artifacts have undergone a transition from instruments for work to media for culture. Design has emerged as a paradigmatic category of informatics.

References

Peter Bøgh Andersen, Beritt Holmqvist, Jens F. Jensen (eds.), (1993). The computer as medium. Cambridge: Cambridge University Press.

Harry Braverman, (1974). Labor and monopoly capital. The degradation of work in the 20th century. New York: Monthly Review Press.

Richard Coyne, (1995). Designing information technology in the postmodern age. Cambridge, MA: MIT Press.

Hubert L. Dreyfus, Stuart E. Dreyfus, (1986). Mind over machine. The power of human intuition and expertise in the era of the computer. New York: The Free Press.

Pelle Ehn, (1988). Work-oriented design of computer artifacts. Stockholm: Arbetslivscentrum.

Joan Greenbaum, Morten Kyng, editors, (1991). Design at work. Cooperative design of computer systems. Hillsdale, NJ: Lawrence Erlbaum.

Paula Elvira Keller, (1998). Arbeiten und Spielen am Arbeitsplatz. Eine Untersuchung am Beispiel von Software-Entwicklung. Frankfurt, New York: Campus.

Brenda Laurel, editor, (1990). The art of human-computer interface design. Reading, MA: Addison-Wesley.

Frieder Nake, (1992). Informatik und die Maschinisierung von Kopfarbeit. In W. Coy, F. Nake, J.-M. Pflüger, A. Rolf, J.

Seetzen, D. Siefkes. R. Stransfeld (eds.): Sichtweisen der Informatik. Braunschweig: Vieweg, 181-201

Frieder Nake, (1994). Human-computer interaction: signs and signals interfacing. Languages of Design 2, 193-205

Bonnie A. Nardi (ed.), (1996). Context and consciousness. Activity theory and human-computer interaction. Cambridge, MA: MIT Press.

Donald A. Norman, (1993). Things that make us smart. Defending human attributes in the age of the machine. Reading, MA: Addison-Wesley.

Heidi Schelhowe, (1997). Das Medium aus der Maschine. Frankfurt: Campus.

Lucy Suchman, (1987). Plans and situated actions. The problem of human machine communication. Cambridge: Cambridge University Press.

Peter Wegner, (1997). Why interaction is more powerful than algorithms. Comm. ACM Vol. 40, No. 5 (May) 80-91

Terry Winograd, Fernando Flores (1986). Understanding computers and cognition. A new foundation for design. Norwood, NJ: Ablex.

Terry Winograd, (1996). Bringing design to software. Reading, MA: Addison-Wesley.

■