DESIGN ARTEFACTS

Towards a design-oriented epistemology

Olav W. Bertelsen

Department of Computer Science, University of Aarhus, Denmark

Abstract

The production of design-oriented knowledge cannot be dealt with fruitfully based on traditional epistemology. In this paper the concept of design artefacts is suggested as a possible backbone in a future design-oriented pragmatic epistemology. It is claimed that a general understanding of how design is mediated is a necessary first step in understanding the use and production of design-oriented knowledge

Keywords:

Design artefacts, epistemology, design, activity theory.

INTRODUCTION

What is the nature of the knowledge needed in designing computer artefacts? How is it possible to assess that knowledge? Most epistemologies, the philosophies of science, are treating science and research as means for gathering true, universal knowledge. It has, however, been argued that the universal criteria for valid scientific conduct have nothing to do with actual knowledge production. For instance, Feyerabend argues that:

> "There is no special method that guarantees success or makes it probable. Scientists do not solve problems because they possess a magic wand -- methodology, or a theory of rationality -- but because they have studied a problem for a long time, because they know the situation fairly well, because they are not too dumb. " (Feyerabend 1975, p. 302)

The idea of a universal methodology becomes particularly problematic in the relation to design oriented disciplines. The epistemologies do not tell us how to deal with the production of design oriented knowledge. Information systems development, human-computer interaction, computer supported cooperative work and other fields related to the design and use of computer artefacts, build on a broad range of disciplines from cultural analysis through programming and hardware construction. Because computer artefacts eventually are to be used in the real world, research in these fields tends to emphasise relevance over scientific rigour. Our field is a hodgepodge, where it is hard to say that one result of research is better than another one; it easily becomes a matter of taste.

> "All methodologies have their limitations and the only `rule' that survives is `anything goes'." (Feyerabend, 1975, p. 296)

The question is whether it is possible to formulate a design-oriented epistemology that acknowledges that "anything goes" in craving for good design and at the same time can be used as a handle in knowledge appraisal. This paper introduces the concept of design artefacts as a generic term for the outcomes of research in design and use of computer artefacts, e.g. theories, methods compilers, editors, debuggers, and case tools. The concept of design artefacts entails concerns for design and use, and the relation to research and method development. The purpose of the concept is to strike a balance between including too much of the richness of reality, and reducing too much, being too concept driven in research. It is argued that this concept contributes to the foundation of the needed design oriented epistemology. It is the claim that an understanding of how design is mediated by design artefacts in general, is the necessary *first* step in developing a design-oriented epistemology.

DESIGN IS A MEDIATED ACTIVITY

In this paper, the design of information systems, and computer artefacts in general, is understood as an activity oriented to changing another activity through the construction and introduction of new (computer) artefacts. In the broadest sense, design is understood as the reformation of conditions for human (working) life. The basic assumption is that *design is a heteropraxial activity*, i.e., involving groups of people with different backgrounds and different motivation for participating in the process. This general heteropraxiality of design can be explained by taking the cooperative prototyping session (Bødker & Grønbæk, 1996) as a *miniature picture* of design in general. Cooperative prototyping sessions, like those reported by e.g. Bødker & Grønbæk (1996), can be interpreted as an extract, exhibiting the basic features of design in general. In this picture, cooperative prototyping is a process where one or more designers work together with one or more prospective users on developing new computer support for the users. Due to differences in professional background, training and interest, the two parties never achieve full explicit understanding of each other. Users do not gain a full technical understanding of the constructed artefact, and the designers, no matter how participatory their approach is, never gain a full understanding of how the artefact is going to be used. However, this is not a problem, because during design users and designers, together, construct a reified, implicit, common understanding in shape of the final prototype. Designers and users act together, but with incommensurable conception of the situation and its aim. In general, many system development projects result in unsatisfying solutions for the users. The successful cases, however, can be attributed to successful acting together without explicit shared understanding.

It is a tenet of activity theory that all human endeavour is mediated by *socially constituted arte-*

facts (Vygotsky, 1978; Leontjev, 1978, 1981; Engeström 1987). Thus, also information systems design is mediated by artefacts. In this paper such artefacts are called design artefacts. In a standard activity theory terminology design can be characterised as an activity where, a designing subject shapes the design object by means of some design artefacts. The design object is the artefact formed in design; the outcome that design activity is directed to. The designing subject is the above-mentioned collection of professional designers and members of the practice the design object is intended to mediate. Design activity is mediated by design artefacts. Examples of design artefacts are programming languages, CASE-tools, specification standards, systems development methods and theories of human motor performance. In cooperative prototyping the prototype plays two roles, it is the continuously moving object of the design activity, but at the same time it is a design artefact mediating the creation of insights and visions into the new information system. As is argued below, this kind of doubleness is a basic feature of information systems design.

Representations play an important role in information systems and in information systems design. Most often information systems can be seen as representations of something, and in most cases design artefacts are themselves representations or they rely on representations. In understanding representations, Wartofsky (1973) outlines a classification of artefacts and how their mediation of human activity is related to representation. In this classification, primary artefacts are those used directly in the production of the means of existence and reproduction of the species. Secondary artefacts are representations used to preserve and transmit productive practice, modes of action, and skills.

> "Secondary artifacts are therefore representations of such modes of action, and in this sense they are mimetic, not simply of the objects of an environment which are of interest or use in this production, but of those objects as they are acted upon, or of the modes of operation or action involving such objects." (Wartofsky, 1973 p.202).

Tertiary artefacts are not related, in a direct sense, to productive practice; they reside in an "off-line loop", where they indirectly influence productive practice by mediating changes in modes of perception and thereby the modes of action. Tertiary artefacts change the way we act by changing the way we *perceive* the world. Wartofsky (ibid.) understands perception as historically developed outward action; perception and action are two sides of the same coin. Based on Wartofsky's notion of primary, secondary, and tertiary artefacts, it is possible to define three dimensions of design mediated by design artefacts; *construction, cooperation*, and *conception*.

Construction

Construction is the productive relation between the designing subject and the object of design. System development is a process aiming at the construction of software and the environment for its use. Seen as construction, design of computer artefacts is a process of programming – an engineering process. The artefacts mediating the relation between designers and the design object seen as technical construction are programming environments, CASE tools etc. Deign artefacts most often mediate all the three dimensions: construction, cooperation and conception. The use of a CASE tool for construction makes the designers conceive the design object with the concepts supported by the tool, and will thus guide the process in certain directions. When looking at design as construction, the advantages of object-oriented programming languages are the possibilities of code reuse, the high degree of maintainability, and the stability of "code structure", (these advantages in a way also relate to cooperation and conception).

Cooperation

Cooperation is the representational relation between subjects involved in design. Design is a cooperative enterprise where different people with different professional backgrounds and different motives are engaged in creating something new, the object of design. Design artefacts mediate system development as cooperation, both as explicit means of cooperation (e.g. status reports and specifications) and as means for the sharing of experiences, insights, and visions about the design object.

A prototype mediates cooperation in a communicative fashion by letting users, through their exploration of the prototype, impose knowledge about their context onto designers, and by letting designers express their new insights by way of continuous change to the prototype.

Object-oriented design artefacts mediate cooperation in another way. The theories of concept building referred to in some of the literature (e.g. Madsen, 1993 chapter 18, Coad & Yourdon, 1990) imply an ontology stating that object-oriented modelling, aggregation, classification etc conform to the basic constitution of the social and natural worlds and with human cognition in general. Activity theory, including Wartofsky's' historical theory of perception, does not agree with this implicit ontology. From this perspective, however, the virtue of object-oriented modelling can be understood in terms of how it mediates the cooperative (and to some extend the conception) dimension of design. It is not the "naturalness" of object-oriented models, but the primitiveness of concept building that supports communication between users and designers around object-oriented models, and the simplicity of software structuring, through modularization etc., that enables users' direct engagement in the constructive dimension through modifying the software.

In agreement with Wartofsky's (op cit.) understanding of representation as outward action, participating in the process of modelling in design can be seen as tightly linked with, and depending on, participation in construction and change. The cooperative dimension depends on representation; not in the naturalistic sense, being true and neutral, but representations supporting specific acts towards the object of design. The secondary artefacts supporting the cooperative dimension, mediate what Strauss et al. (1985) refer to as articulation work, "the specific details of putting together tasks, task sequences, task clusters, and even the work done in aligning larger units such as sub projects, in order to accomplish the work" (p. 175).

Conception

Conception is the dialectical relation between the designing subjects and the historically developing activity. Design is conception of the considered practice and of the new to come. Users and designers achieve new understanding of the existing world and rooted in this they conceive a new world that transcends the old. Design artefacts mediate learning and conception. A functional programming language for example facilitates the understanding of the design object as transformation of data streams. Conception is about changing the existing modus operandi, both by identifying more suitable ways of realising the established activity and by developing entirely new motives. Conception is both understanding (of the given) and imagination (of the better). A prototype induces new visions and knowledge when it confronts the possible to the existing. Mogensen (1994) has in detail explained how prototypes can be used as an analytical device. His idea is that by introducing a new artefact into a given practice, constrains and potentials become explicit. In the present terminology this means that a prototype mediates conception.

Any design artefact, no matter what its intended use is, to some extend mediates all the three dimensions of design. A debugger mediates construction by providing efficient feedback when the programmer is coding; it mediates cooperation by representing programs and bugs in a standardised way, thus facilitating programmers joint work on the program; it mediates conception by yielding alternative views on the actual program and serving as a way to learn about other programmers' code, e.g. in standard libraries. In the same way, a system description mediates construction by being a high level programming tool; it mediates cooperation by fixing ideas to be shared and by being a vehicle for the division of labour; it mediates conception by inducing alternative, structured views into the problem in question.

The next four sections discuss how design artefacts mediate across heterogeneity; how design artefacts have to mediate across the contradiction between construction and representation; how design artefacts do not prescribe design; and how theories are design artefacts. In the final section, it is discussed how the concept of design artefacts can be the basis for a design-oriented epistemology.

DESIGN ARTEFACTS MEDIATE ACROSS HETEROGENEITY

Design artefacts mediate across heterogeneity in that they tie involved activities together, and in that they tie different rooms, phases or zones, of design and use together.

The heterogeneous activities contributing to design are tied together through their joint use of artefacts and through their joint focus on the same object. Users and designers are driven by different motives and the object of design does not make sense in the same way for them. Design artefacts tie different communities of practice together, maintaining meaning across groups but making sense in different ways to the different groups. When designers and users work together on a design specification, designers may perceive the specification as a model of the data flowing in the prospective system, whereas users may understand the same specification in terms of a working procedures in the organisation. Design artefacts do not only take different shapes and serve different purposes to different groups, they also take different functions within one group across time, during use and design, and in the different rooms (or phases) of a specific group's practice. In this way, system development method has one function in the project organisation's internal education prior to a project, and another function during the project. Working in accordance with the method means two different things in the two rooms, or phases, of a project.

Using Star's (1989) terminology, design artefacts are boundary objects because they adapt to different situations of application and at the same time maintain identity, thereby mediating divergent needs and viewpoints. As illustrated by using the cooperative prototyping session as a general miniature picture of design, also the object of design is a boundary object. This boundary objectness reifies shared understanding in design. Hence explicitly formulated shared understanding may not be needed as long as design artefacts support the making of the design objects that fit all the incommensurable communities of users, designers, etc.

Following the emphasis on heterogeneity, design can be described as a process of *transforming artefacts* from the domain of use. In such a process, design artefacts mediate across use and design; designers and users; and across construction and representation. Bærentsen (1989) has explained that development of technology can be understood as crystallisation of activity with one generation of an artefact into artefacts in the next generation. In the case of weapon technology, Bærentsen's example, the flint gun crystallises the manual act of lighting the gunpowder when using earlier generations of handguns. Understanding design as transformation is to understand it as a process of deliberate crystallisation. Hence, design is to take artefacts from the considered practice into the room, or zone, of design. In this room the artefacts are changed and eventually they are brought back into the practice of use. The success of the transformed artefacts depends on their ability to mediate use when they have been brought back. This ability may be ensured by continuously confronting the artefacts under transformation with (simulated) use.

An example of design understood in terms of transformation comes from a project with a large music festival (Bertelsen 1996, 1998). To support their own work, one group of festival volunteers developed a checklist, crystallising established routines. The checklist was later spread out to larger parts of the organisation. Even later, the checklist entered a design project developing support for festival planning, it became a database sketch, an object oriented model and finally a prototype of a planning tool. The final system was a computer-based version of the checklist, being able to print out "checklists" on papers. Bertelsen (1996) concludes that the lack of commitment in the festival organisation, to enable serious confrontation with simulated use during the design process was the main reason for the failing success of the final system.

The transformation of artefacts through design is a representation process. The transformed artefacts both exist in their original form within the room — or domain — of use, and as representations in the room of design. Hence, artefacts being transformed are boundary objects (Star, 1989) in the sense that they exist and mediate across different activity systems, across groups, across phases of practice, and across the transition between use and design. Understanding design as the transformation of artefacts emphasises concreteness of representations in design; thereby it contradicts viewing representations in design as the abstract unfolding of general constitution of the social and technical domain. Examples of practical techniques supporting the concreteness of representations can be found in the participatory design literature, e.g. cooperative prototyping (Bødker & Grønbæk, 1991), organisational games (Ehn & Sjögren, 1991), and mock-ups (Ehn & Kyng 1991).

A design-oriented epistemology should address the important issue of mediation across heterogeneity. Firstly, by supporting that design artefacts, the outcomes of design-oriented research, are suited for mediating the three dimensions of design, construction, cooperation and conception, and by ensuring that design artefacts make sense in different rooms of (design) practice. Secondly, the need for mediation across fundamentally different rooms of use, design and research, implies an emphasis on modes of representation that take development and change as basic features and that are supplying handles for the designer in handling such dynamics

REPRESENTATION VERSUS CONSTRUCTION

Design artefacts must mediate a *basic contradiction between representation and technical construction*. This double character of being split between representation and technical construction is prominent in relation to many aspects of design. Technical construction of computer systems is extremely formalised, whereas representations in design depend on a high degree of openness and interpretability. This double character is a problem because the plasticity of secondary artefacts is obstructing or obstructed by the naturalism and formalisation of primary artefactness. The example below illustrates this requirement for design artefacts.

A small in-house development department experimented with using the diagrams from a CASE tool, based on structured design (Yourdon 1982), together with users. They found that, output from the CASE-tool both facilitated and obstructed user-designer cooperation. In one particular situation, a context diagram (poetically referred to as the "the sunflower") was successfully applied as a common frame of reference shared by the whole project group during the project. In another situation the designers had no luck trying to apply diagrams from the same CASE-tool together with the users in the project group (Laursen et al. 1990).

From a technical perspective, structured design (Yourdon 1982) is a high-level program construction tool, but at the same time this design artefact mediates cooperation and conception. In both the above situations, the CASE tool and its output had a double role, both mediating production of machine executable code and mediating cooperation and the conception of the new system. In the first situation (the sunflower) the technical contents was simple enough to allow for the project groups free attribution of meaning to the diagram, whereas in the other situation the model was so complicated and incomprehensible for the users, that it was impossible to do anything with it beyond trying to understand what it was about. The "sunflower" offered openings into a poetic world, whereas other diagrams only yielded frustration. The applicability of structured design tools in cooperation and conception with users depends on the designers' success in establishing common representations based on the tool.

A distinction between meaning and sense, similar to the distinctions in linguistics between what a sign points to and what it does in use, can be applied in analysing design artefacts. Technical instruments have both meaning and sense in the following way. The meaning of a hammer is the assemblage of shaped iron and wood. The sense of the hammer is that it is used by carpenters for driving nails. The sense of the hammer is "crystallised" in secondary artefacts, mediating hammering practice; e.g. anecdotes and sayings. The meaning of the "Sunflower" described above was that it was a (formalised) SA/SD context diagram. The sense of the "Sunflower" in the cooperation between designers and users was the context of the prospective computer system, and the way it would mediate parts of the relations between activities in the organisation. This has some degree of tertiary artefactness to it, because the poetic connotations of summer and nature may have mediated the emergence of entirely new ways of understanding the workplace.

Programming shares the same kind of doubleness, being both interpretable representation and hardwire construction. Naur (1985) rejects the idea that programming can be seen as the mere production of machine executable code — isolated construction without elements of conception and cooperation. He explains that a theory about what the program does, and how it does it, is built simultaneously with the construction of the executable code. Thus, in the terminology introduced above, programming work has the double character of being both construction and conception. This theory cannot be written down or otherwise formalised, and is only accessible to the programmers working on the particular project. In Wartofsky's terminology Naur's "program theory" is a secondary artefact conserving the acquired knowledge and skills in working with the program. However, Naur bases his analysis on the individualist philosophy of Ryle, thus neglecting the societal/cultural aspect of representation. The concept of secondary artefacts helps us understand that the way humans understand their surroundings (including programs) is culturally mediated. Secondary artefacts not only conserve knowledge and skill among the individuals whose experience they are based on, but secondary artefacts also transfer these across a given culture, e.g. the programming profession.

A design-oriented epistemology should explicitly addressed the contradiction between representation and construction, thereby providing means for handling the transition from interpretation to running software. Furthermore, such an epistemology should support the production of knowledge about the social nature of technical issues, e.g. about the cultural aspects of source code indentation.

DESIGN ARTEFACTS DO NOT PRESCRIBE PRACTICE

Design artefacts do not prescribe design practice. In most cases a specific design artefact is intended to be used in a specific way, and just as often the actual way in which the design artefact mediates design is very different from this intention. In most cases, the relation between the principles of intended use and the actual practice with the artefact is far from trivial. This discrepancy is most obvious in the case of design methods, illustrated by the below excerpt (reconstructed from interview notes) from dialog between an interviewer (I) and a designer (D) of embedded software.

I: How do you do when you specify the software for the pumps?

D: Well, according to the SPU method … [lengthy summary of a method book].

I: OK. The pump we looked at in the front room, the one you designed the software for, how did you make the specification for that?

D: With that particular pump the situation was a little unusual, so we did not go through the steps of the SPU method.

[…]

I: Let's talk about the project you are currently involved with. How is specification done there?

D: The current project is not a normal design project, because we are also developing a general framework.

The embedded software department at the machine factory had used some effort in implementing the SPU method (Biering-Sørensen, et al., 1988), but the interviewed designer had never gone through the steps, defined in the method book, in a project. This is in accordance with findings of Button & Sharrock (1994), who studied the use of design methods in a development project and found a tension between methodologists' ambition (clarity, coherence and closure) and the indeterminate character of actual design work. This mismatch between the prescriptions of the method and the actual design work can be understood in terms of the doubleness of meaning and sense of the method. The meaning of the method being the literal prescription of how to proceed through specified steps, filling in certain forms etc.; the sense of the method being the actual use of it, the sparse use of selected techniques combined with common sense, etc. However, when the distance between this "meaning" and "sense" becomes too long, the meaning-sense relation degenerates to a contradiction between principles and practice.

The general discrepancy in design artefacts between what they prescribe and the practice they induce can in some situations be understood as a contradiction between the design artefact producing activity and design practice. In other situations, the discrepancy between principles and practice can be understood as a result of the relation between meaning and sense in design artefacts. In the case of the designer of embedded software, the meaning of the SPU method was the prescriptions for design printed in the method book, whereas the sense of the method was the way it actually influenced work in concrete design projects. Methods, or method books, are secondary artefacts in the sense that their prescriptions are rooted in practice. But, following Wartofsky's argument, secondary artefacts are not transferring and maintaining modes of action because of their naturalism. Some method books are based on fiction, war stories from other fields and the like, which in relation to the purpose of a method book has a character of tertiary artefactness, reshaping the designers' view on their own practice. We cannot expect methods to prescribe practice in a direct manner. Instead, the prescriptions embedded or expressed in design artefacts are resources; they do not determine process and product; they are seeds that can develop in concrete practice in varying ways depending on specific circumstances.

A design-oriented epistemology should reflect that knowledge, no matter whether it has the form of theory, method or tool, is used in unforeseen ways. Because knowledge about context and scope is often left out when the outcomes of research are applied (Meister & Farr, 1967), this exact knowledge should be made an operational part of the design-oriented knowledge, the design artefacts. However, since the outcomes of design-oriented research should not only be considered as something conveying universal knowledge and universal prescriptions for design, the outcomes of design-oriented research can never become completely self-contained and portable. The next section explores how theories can be understood as design artefacts.

THEORIES ARE DESIGN ARTEFACTS

Theories are design artefacts. The outcome of research is often formulated as theory; that is, highly universal representations, pretending to be independent of concrete context, practice or situation. From the point of view of design, theories are design artefacts, taking different roles in design; from worldviews, guiding the designer and helping him assess the situations and keep the goals in mind, to tools mediating the achievement of specific results. In this way, the direct outcomes of research mediate design.

The application of Fitts' law in human-computer interaction illustrates how theories act as design artefacts. Fitts' law (Fitts, 1954) describes the limitations of human motor performance, as established and verified through laboratory experiments. The law is based on mathematical information theory (Shannon & Weawer, 1949), and states that there is a specific relation between the length of a movement, the size of the target and the time it takes. Fitts' law has been used extensively in human-computer interaction research, often in analysing novel input devices. Based on the application of Fitts' law in the human-computer interaction literature, Bertelsen (1994) describes three roles this specific theory takes as a design artefact: worldview, tool for calculation, and metaphor. The calculation role has to do with the construction dimension of design, by supporting interface optimisation in well-defined ways. The metaphor role is related to cooperation by supplying a vocabulary for communication about pointing and screen layout issues' effect on pointing time, and it is related to conception by setting a frame of thought in the conception of new interface ideas. The worldview role is a matter of supporting conception. In this role, Fitts' law expresses the basic view on human-computer interaction as being decomposable and possible to treat through laboratory experiments. Like any other design artefact, Fitts' law most often take more than one role at a time. By thinking about interface problems in Fitts' law terms, the metaphor role, other views are excluded. Fitts' law then draws the main view on the design task towards time performance, becoming a Trojan horse for a cognitivist worldview. The three roles of Fitts' law as a design artefact, described by Bertelsen (1994), cannot be generalised. However, they span a spectrum of ways in which theories mediate design. Between being tool, which is goal directed and "neutral", and being worldview which is value based and motive oriented. Mediating construction, cooperation and conception.

As another example, activity theory spans a similar range of roles as a design artefact. It takes a tool-like role, when used as the basis for analysing empirical data of interaction with a specific computer artefact in terms of focus shifts (Bødker 1991). It takes a worldview-like role when, designers adopt the activity theory framework as their way of understanding technology, i.e. as culturally constituted mediation of human activity etc. It takes a more metaphor-like role, as a vehicle for communication, in shape of the well-known double triangle figure (Engeström 1987) or as activity analysis (Korpela et al., this issue). While, activity theory mediates cooperation and conception in quite clear ways, it is still an open question how it may mediate construction in a direct sense.

Theories mediate construction, communication and conception by playing different roles in the continuum from tool to worldview, being primary, secondary and tertiary artefacts. Theories explicate the doubleness of design artefacts of mediating construction and representation; as in the case with Fitts' law both mediating technical optimisation of user interface layout, and representing specific understandings of the interface design problem. Theories, like other design artefacts, are boundary objects, most prominently by tying the production of "universal facts" in research together with design. The notion of theories as design artefacts contributes to a pragmatic philosophy of design science by providing a criterion for the assessment of theory.

A design-oriented epistemology should acknowledge that theories are design artefacts mediating construction, cooperation and conception.

TOWARDS A DESIGN-ORIENTED EPISTEMOLOGY

Above, the concept of design artefacts has been outlined. Through the explicit focus on heterogeneity and socially developed mediation, this concept turns out to be a fruitful vehicle in understanding knowledge production related to the design of information systems. The paper has pointed to specific features of design artefacts and their implication for a design-oriented epistemology.

First of all, by understanding theories as being design artefacts, traditional concerns over method, rigor and so forth— concerns that are hard to deal with across the broad range of disciplines involved in the design of information systems—become easier to handle, in that they are balanced against the concern for practical relevance. The test bed for theories related to the development of information technology is not the controlled scientific laboratory, but the utilisation in practical design. This does not mean, however, that anything goes. It only points out the direction for a new kind of rigor in information systems research, rejecting irrelevant results, and acknowledging the particular and the situation specific.

Design artefacts simultaneous mediation of construction, cooperation and conception implies that the production of design-oriented knowledge is subjected to a broad range of criteria for validity. Knowledge valid for construction may be produced according to methods emphasising predictability and generality. Knowledge valid for cooperation must include a far broader range of criteria focusing on interpretation and collective action. Knowledge relevant for conception must be produced according to criteria and methodologies that enables research to acknowledge transitions between the present the past and the future; methodologies that does not require detachment form context and isolation in time. In short, development and change must be acknowledged as a basic feature of reality — conveniently, this acknowledgement is a basic tenet of activity theory. More specifically, design as transformation or artefacts is an example of a basic concept acknowledging change and development as a basic feature of the object of information systems research

Design artefacts mediate across the heterogeneous rooms of use, design and research, and across the multiple dimensions of design. The concept of design artefacts contributes to an understanding of the mechanisms underlying the intertwinement of different groups and professions in the networks of use, design and research activities, by maintaining focus on material mediation. The concept of boundary objects has been extended to denote mediation across situations, and not only across heterogeneous groups. Crystallisation and design as transformation of artefacts are mechanisms build on this extended meaning of boundary objectness across historical generations of the same practice. Understanding how design artefacts are extended boundary objects, may be a resource in understanding how knowledge can be transferred, or transformed, from research to practice and vice versa.

The doubleness in the contradiction between representation and construction in design is a problem that a design-oriented epistemology should address explicitly. Design-oriented knowledge, taking the form of theories, tools, or methods (or something else) should include, or be accompanied by directions of how to deal with the transition from the multiple voices of interpretation to the hardwired meaning of running program code.

Finally, a design-oriented epistemology should acknowledge that design artefacts do not prescribe actual design practice. This gives research and method development the freedom of not needing to encompass the entire reality. The production of design-oriented knowledge can afford a high degree of rigidity and narrow-mindedness, designers always adapt methods, tools and theories to actual practice.

The concept of design artefacts offers a vehicle for general self-reflection in design-oriented research, and for understanding the role of theory in particular. In terms of an epistemology, the concept of design artefacts implies that validity is a partial aspect of theory and research. An epistemology based on the design artefact concept enables and accepts an incoherent collection of partial theories as a valid "science base" for design without turning into relativism. Hence, negating the tenet of logical positivism that theory should be universally applicable independently of context and purpose; always being true and independent of motive and values. The concept of design artefacts unifies concerns related to use, design and research in information systems in a pragmatic, design oriented epistemology.

> "At all times man approached his surroundings with wide open senses and fertile intelligence, at all times he made incredible discoveries, at all times we can learn from his ideas." (Feyerabend 1975, p.307).

Acknowledgements

Susanne Bødker, Kari Kuutti, Kim Halskov Madsen, and Marianne Graves Petersen have supplied fruitful criticism of this paper at various stages of the writing process. The paper is based on work funded by The Danish Research Programme for Informatics, grant number 5.26.18.19, DEVISE, and The Danish Basic Research Foundation, Centre for Human-Machine Interaction.

References

Bertelsen, O. W. (1994). Fitts' Law as a Design Artefact: A Paradigm Case of Theory in Software Design. In Blumenthal, Gornostaev, & Unger (eds.). *Human-Computer Interaction. 4th. International Conference, EWHCI '94 St. Petersburg, Russia, August 1994. Selected Papers*. Berlin: Springer Verlag. pp. 11-18.

Bertelsen, O. W. (1996). The Festival Checklist: design as the transformation of artefacts. In Blomberg, J., Kensing, F. & Dykstra-Erickson (eds.). *PDC '96, Proceedings of the Participatory Design Conference*. Palo Alto: Computer Professionals for Social Responsibility. pp. 93-101.

Bertelsen, O. W. (1998). *Elements to a theory of design artefacts: a contribution to critical systems development research*, Ph.D. Thesis, Aarhus University. DAIMI PB531. (available from http://www.daimi.au.dk/~olavb)

Biering-Sørensen, S., et al. (1988). *Håndbog i struktureret programudvikling*. [Handbook of structured program development]. Copenhagen: Teknisk Forlag.

Bærentsen, K. (1989). Menneske og maskine [Man and Machine]. In Hedegaard, Hansen & Thyssen (eds.). *Et virksomt liv* [An active life]. Aarhus: Aarhus University Press. pp. 142-187.

Bødker, S & K. Grønbæk (1991), Design in Action: From Prototyping by Demonstration to Cooperative Prototyping. In Greenbaum, J. & M. Kyng (eds.). *Design at work: cooperative design of computer systems*. Hillsdale: LEA. pp. 197-218.

Bødker, S. & Grønbæk, K. (1996). Users and designers in mutual activity: An analysis of cooperative activities in system design. In Engeström, Y. & Middleton, D. (eds.). *Cognition and Communication at Work*. Cambridge: Cambridge University Press. pp. 130-158.

Bødker, S. (1991). *Through the Interface: a human activity approach to user interface design*. Hillsdale, N.J.: LEA.

Button, G. & Sharrock, W. (1994). Occasioned practices in the work of software engineers. In Jirotka, M. & Goguen, J. (eds.). *Requirements Engineering Social and Technical Issues*. London: Academic Press. pp. 217-240.

Coad, P. & E. Yourdon (1990). *Object-Oriented Analysis*, Engelwood Cliffs N.Y.

Ehn, P & M. Kyng (1991). Cardboard Computers: Mocking-it-up or Hands-on the Future. In Greenbaum, J. & M. Kyng (eds.). *Design at work: cooperative design of computer systems*. Hillsdale: LEA. pp. 169-198.

Ehn, P. & D. Sjögren (1991). From system description to script for action. In Greenbaum, J. & M. Kyng (eds.). *Design at work: cooperative design of computer systems*. Hillsdale: LEA. pp. 241-268.

Engeström, Y. (1987). *Learning by expanding: an activity-theoretical approach to developmental research*. Helsinki: Orienta-Konsultit Oy.

Feyerabend, Paul (1975). *Against Method*, London: New Left Books.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. In *Journal of Experimental Psychology* vol. 47, no. 6. pp. 381-391.

Korpela, M., H. A. Soriyan & K. C. Olufokunbi (this issue). Activity analysis as a method for information systems development: General introduction and experiments from Nigeria and Finland. In *Scandinavian Journal of Information Systems, vol. 12, 2000.*

Laursen, B, et al. (1990). *Dokumentation & Kommunikation i oliebudgetprojektet*, [Documentation and Communication in the oil-budgeting project.]. unpublished case study report. Department of Computer Science, University of Aarhus.

Leontjev, A. N. (1978). *Activity, consciousness, and personality*. Engelwood Cliffs NJ: Prentice Hall.

Leontjev, A. N. (1981). *Problems of the development of the mind*. Moscow: Progress.

Madsen, O. L., Møller-Pedersen, B. & Nygaard, K. (1993). *Object-Oriented Programming in the BETA Programming Language*. Wokingham, England: ACM Press/Addison-Wesley Publishing Company.

Meister, D. & Farr, D. E. (1967). The Utilization of Human Factors Information by Designers. In *Human Factors*, vol. 9. pp. 71-87.

Mogensen, P. (1994). *Challenging Practice: an Approach to Cooperative Analysis*, Ph.D. thesis, Aarhus University: DAIMI PB-465.

Naur, P. (1985). Programming as theory building. In *Microprocessing and Microprogramming* vol. 15, pp. 253-261.

Shannon, C. E. & Weawer, W. (1949). *The mathematical theory of communication.* Illinois.

Star, S. L. (1989). The Structure of Ill-Structured Solutions: Boundary Objects and Heterogeneous Distributed Problem Solving. In Gasser, L. & M. N. Huhns *Distributed Artificial intelligence, volume II*. London: Pitman Publishers. pp. 37-54.

Strauss, A., Fagerhaug, S., Suczec, B., & Wiener, C. (1985). *Social organization of medical work.* Chicago: The University of Chicago Press.

Vygotsky, L. (1978). *Mind in society: The development of higher mental processes*. Cambridge, MA: Harvard University Press.

Wartofsky, M. W. (1973). Perception, representation, and the forms of action: toward an historical epistemology. In Wartofsky, M. W., *Models*. Dordrecht: D. Reidel Publishing Company, 1979. pp. 188-210.

Yourdon, E. (1982). *Managing the system life cycle*. New York: Yourdon Press.