Understanding Design

Gerhard Fischer and Heinz-Dieter Boecker
Research Group on Man-Machine Communication
Computer Science Department, University of Stuttgart

Abstract

Design is concerned with how things ("artifacts") ought to be in order to attain goals and function. To gain a deeper understanding of the nature of design, we have studied design processes in several different domains: software engineering, creative writing, building models with technical kits and constructing learning environments.

1. The nature of the design process

Designers try to shape the components of new structures. In general the designer is facing an ill-structured problem in a semantically rich domain. An emerging design is incorporated in a set of internal and external memory structures (eg on paper, in computer memory). The problem solving process for this task can best be characterized as a dialectical information gathering process switching back and forth between **creating** and **understanding** subprocesses. At each stage in the design process, the partial design (reflected in the descriptive structures) serves as a major stepping stone and as a stimulus for suggesting what should be done next (see ALEXANDER 1964 and SIMON 1969).

There is no doubt that there is domain specific design knowledge: an architect knows a lot about building materials and how to combine them, a software engineer knows many basic algorithms and how to use them in constructing software components. The more interesting question in research about design is whether there is also some general design knowledge, which is applicable in many different domains.

2. Universal concepts for design processes

Crucial processes in **creating a design** are:

1) to develop a language for specifying goals and a set of intermediate concepts (eg in the case of technical construction systems: a steering unit, a differential)
2) to deal with sets of possible worlds (ie we have to consider design alternatives)
3) to incorporate the emerging design in a set of external memory structures
4) to record the relevant parts of the design path (and not only the design product)
5) to create low-cost modifiable models which help us to replace anticipation by analysis and experimentation

Understanding (analysis) of existing designs is relevant for the following reasons (especially if we regard the design primarily as an error correcting process):

1) it is a prerequisite for modification (which is necessary because it is not possible for the designer nor for the potential user to foresee all of the opportunities for which a complex system is used).
2) the insufficiencies of existing prototype models have to be uncovered; the surface manifestations of problems have to be traced back to their causes.
3) contemplating and experimenting with what has been done so far can stimulate new ideas, new generalisations, new simplifications and new uses for developed parts. New possibilities may become visible through perception of unplanned relationships.

Computer science has developed a set of well known (eg top-down, bottom-up, stepwise refinement) and less known (eg use of stepping stones, progressive constraints, structured growth, linguistic uniformity, uniformity of interaction, metadescription, context-sensitive descriptions, filters) concepts of design.

3. Technical construction systems and systems for man-machine communication

To gain a better understanding for an epistemology of design, we have studied design processes in several domains: software engineering (BOECKER, FISCHER and GUNZENHAEUSER 1980), creative writing (FISCHER 1980), technical construction systems (FISCHER and BOECKER 1981) and learning environments (FISCHER, BROWN and BURTON 1978). We have verified and extended the basic set of thought processes underlying design and the organization of those processes which serve as guidelines for the implementation of design support systems.

In our own work we have chosen a specific technical construction system, namely **FISCHERTECHNIK (FT)**. FT can be enjoyed by "children of all ages". It allows the designer to construct starting with a universal building block - very simple models (eg a car, a crane etc) as well as completely functional models (eg a technically exact model of an assembly line for car manufacturers). FT is not only an excellent toy (it gained the "Oscar de Jouet" in 1972 for being elected the best toy of the year) but a realistic design tool which allows to build artifacts of comparable complexity relative to big software systems.

FT is a good example of a design tool that has the property **"no treshold, no ceiling":** easy things are easy to do and difficult things are possible to do. The most important feature of FT is probably the mechanism to join individual building blocks (which is uniform and universal and can be undone in all cases). It provides the basis for an environment which allow us to study sophisticated design processes with great similarity to software engineering.

The other major research area in our work for an epistemology of design has been the **design of man-machine communication systems.** We will make use of the substantial computational resources at the hardware level (which will become available in the near future) to create a symbiotic environment for man and machine which combines the advantages of both to cooperatively solve design tasks which neither of them could solve alone. The main difficulty in this domain is our lack of a theory which is predictive enough to provide a complete set of design criteria. We try to make use of the knowledge about the human information processing system (eg limited STM, good visual chunking methods) to come up with some principles (eg consistency and adequate default assumptions) for the design of these systems. The few derivations we can make from these principles, however, immediately lead to serious design conflicts. To point out a specific example: if we want to delete a file (out of a large set of successive versions), the default option should be to delete the oldest version whereas if we print or edit a file the default should be the newest version. At the surface level this is obviously an inconsistency. Only if the user has an explicit conceptual model of these processes (eg about the semantics of "delete", "print" and "edit") this seems to be the obvious way.

One of the long range goals of our research is to construct computer-based **convivial tools** (ILLICH 1973). These are tools that give the user as much control as possible so that he can do what ever he wants to do. We consider Fischertechnik a good example of a convivial tool (eg compared with a plastic car which can never be modified).

Our **empirical research** verified some of our intuitions, namely

1) there exist common features for design processes in different domains
2) design is an incremental activity; a partially completed product is important and useful to gain a deeper understanding of the problem;
3) the major difficulty in design tasks is not to understand the intrinsic semantics of individual building blocks but to predict how an assemblage of such components will behave
4) it is important for the understanding of an artifact that we have access not only to the final product but to the important parts of the evolutionary path which led to it
5) the language developed in Computer Science and specifically in AI is adequate for the description of design processes in different domains.

4. A computer system to support design processes

Many design problems (including social planning, VLSI design, software engineering) have demonstrated to us that there are inherent limitations to the complexity that the unaided designer can control in any situation. Our system INFORM can best be described as an integrated, knowledge-based design environment (primarily used for software creation, maintenance and modification) using multiple windows to provide uniform, direct interactive facilities to the user. We expect that systems of this sort (see also GOLDSTEIN and BOBROW, 1980) will make major contributions to cope with the problems of constructing large software products by keeping the whole programming process in the machine (and not only the result) and by allowing to extend our specification techniques with prototyping.

5. Conclusions

Our goals doing this research were: to gain a better understanding of design (a cognitive science task) and to use this understanding to build systems which suport the design process (a cognitive engineering task). We believe that domain independent design knowledge and skills exist and that design concepts are the right vocabulary to describe complicated processes in problem solving, expert systems research and software engineering.

References

Alexander, C. (1964): "The synthesis of form", Harvard University Press

Boecker, H.-D., G. Fischer and R. Gunzenhaeuser (1980): "Die Funktion von integrierten Informationsmanipulationssystemen in der Mensch-Maschine Kommunikation: Projekt INFORM", MMK Memo, Institut fuer Informatik, Universitaet Stuttgart

Fischer, G. (1980): "Integrated Information Manipulation Systems -- A cognitive view", Proceedings of Coling 1980, the 8th International Conference on Computational Linguistics, Tokyo

Fischer, G. and H.-D. Boecker: "The nature of design processes and how computer systems can support them", submitted to IJCAI 81

Fischer, G., J.S. Brown and R. Burton (1978): "Aspects of a theory of simplification, debugging and coaching", in Proceedings of the 2nd National Conference of the Canadian Society for Computational Studies of Intelligence, Toronto, pp 139-145

Goldstein, I. and D. G. Bobrow (1980): "Descriptions for a programming environment", Proceedings of AAAI, pp187-189, Stanford

Illich, I. (1973): "Tools for Conviviality", Perennial Library, Harper and Row, New York

Simon, H. (1969): "The sciences of the artificial", especially chapter 3: "The Science of Design", MIT Press, Cambridge, MA