The Problem of Design Problems

Kees Dorst

In this paper we will concentrate on exploring a subject that has been much neglected in Design Methodology: the structure of design problems. First we describe what we can find in Design Methodology about the structure of design problems. This will focus on the descriptions of design problems as 'underdetermined problems' and on the way design problems are treated within the two paradigms of design methodology. Then we set out to develop a novel approach to the study of design problems, that is based on the situated nature of design problems and the evolutionary nature of design processes.
In doing this, we find that how designers perceive, interpret, structure and solve design problems cannot really be understood without taking their level of design expertise into account.

D esign Methodology has always had something of a blind spot for design problems: the focus in Design Methodology has almost exclusively been on the support of the *process* of designing. But any method for aiding design activities necessarily contains statements or assumptions about all three 'dimensions of design activities' (see figure 1, and (Roozenburg and Cross, 1991)): the dynamics of a design process, the designer and the design problem.
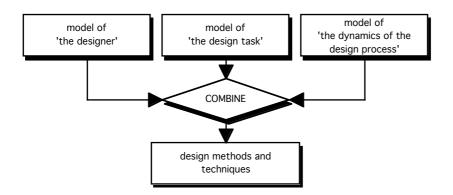
Figure 1 - The three dimensions of design activities.

Within Design Methodology, the nature of design problems has been described as 'ill-structured', or even 'wicked' (Cross 1984), but little more has been said. The work that has been done on design problems is comparatively old, starting straightaway with the early work of Christopher Alexander (1962). Since the 80's, the stream of papers that deal with this have quieted down to a trickle. For an overview of the early work see Cross (1984).

But the study of design problems is still as relevant as ever: most process-focussed design methods seem to incorporate strong assumptions about what design problems are (e.g. concerning the independence of subproblems, the objectivity of problems, the possibility to create an overview of a design problem, etc). To determine the merits and the scope of application of these methods we will need to know more about the structure of design problems (We can only make sure that a design methods is validly applied by comparing the assumptions about the structure of design problems that underlie a design method with the structure of design problems in that particular area of design). Knowledge about the structure of design problems is also important for deepening our understanding of design - if we can find a way to trace the structure of design problems, and we can then match that to the way designers tackle those problems, this will open the possibility for a much closer description and a much better understanding of the way designers work, and WHY they take the actions we see.

An initial theoretical investigation into design problems focuses on the questions: (1) What can we find in Design Methodology literature in terms of descriptions of the structure of design problems? And: (2) How can we use these descriptions to develop a taxonomy of design problems?

1. Design problems

We will start out the exploration by mapping what we know of design problems. In the next sections we will focus on the descriptions of design problems as 'underdetermined problems', and on the way design problems are treated within the two paradigms that underlie Design Methodology.

1.1 Design problems as underdetermined problems

Design activities can be seen as the reasoning from a set of needs, requirements and intentions to a new bit of reality, consisting of a (physical) structure and an intended use. This process of reasoning is non-deductive: there is no closed pattern of reasoning to connect the needs, requirements and intentions with a form of an artifact and a mode of use. This 'openness' of a design problem is called the *underdetermination* of design problems. Upon closer inspection we can actually distinguish two ways in which a design problem is underdetermined: (1) a description in terms of needs, requirements and intentions can never be *complete* (there can never be enough to define a form (Roozenburg and Eekels 1995) and (2) 'needs, requirements and intentions' and 'structure' *belong to different conceptual worlds* (Meijers 2000).

These two kinds of underdetermination together make up an impressive rift between 'the design problem' and ''the design solution'. This may lead to the feeling that the solution of design problems must be well nigh impossible. But designers somehow overcome the underdetermination of design problems and the conceptual rift between 'need' and 'form' in their

design processes. This is a gradual process, involving many steps, that are performed in patterns ('design strategies').

One of the problems designers face in doing this, is that design problems are not completely determined, but also not completely free. Most design problems in fact seem to have a threefold nature:

(1) They are partly **determined** by 'hard' (unalterable) needs, requirements and intentions.   A designer will have to reserve time in the early part of his design process to unearth these 'hard facts' by information gathering and analysis, and live with these specifications. This information can be seen as a necessary input at the start of the design porcess, and this type of interaction can very well be described and modeled within the rational problem solving paradigm.

(2) But a major part of the design problem is **underdetermined**. The interpretation of the design problem and the creation and selection of possible suitable solutions can only be decided during the design process on the basis of proposals made by the designer. These proposals thus entail both the possible interpretations of the design problem and possible solutions to those problems.

(3) Part of the design problem can be considered **undetermined**, in the sense that the designer is to a large extent free to design according to his own taste, style and abilities. (It is of course not the case that the designer would never have to defend these aspects of the design to others, but in these areas the designer is dominant in the sense that he also provides the criteria on which this aspect or part of the design is to be judged).

1.2 Design problems in the rational problem solving paradigm

To describe the ways in which design methodology has treated the whole issue of design problems we have to distinguish between the two fundamentally different paradigms that the field is based on.

The main paradigm of design methodology, in which design is seen as a *rational problem solving* process, was introduced by Simon in the early 1970s. In this paradigm, design is viewed as a rational search process: the design problem defines the 'problem space' that has to be surveyed in search of a design solution. Problem solving theory is concerned with the ways in which people or artificial systems arrive at solutions to problems they encounter. This theory can be captured by four propositions:

> • A few gross characteristics of the human Information Processing System are invariant over task and problem solver
> • These characteristics are sufficient to determine that a task environment is represented as a problem space, and that problem solving takes place in a problem space
> • The structure of the task environment determines the possible structures of the problem space.
> • The structure of the problem space determines the possible programs that can be used for problem solving. (Newell and Simon, 1972)

If this theory is valid for design, design problem solving also takes place within a problem space that is structured by the structure of the task

environment, which in its turn determines the 'programs' (strategies or methods) that can be used for designing.

In a later paper Simon (Simon 1973) addressed some of the difficulties that might arise in applying the rational problem solving approach to design by defining design problems as 'ill-structured problems'. In ill-structured problems the problem space is deemed too large, ill-structured and ill-defined to be described, which means that the possible solutions cannot be enumerated. Ill-structured problems are therefore to be tackled in an 'immediate problem space'. This is a part of the total problem space. The immediate problem space is addressed and put together by an (unspecified) 'noticing and evoking mechanism'. The basic 'design' problem-solving process would however be a rational search process, basically the same as in other kinds of problem solving.

1.3 Design problems in the reflective practice paradigm

A radically different paradigm was proposed fifteen years later, by Donald Schön (Schön 1983), who describes design as an activity involving *reflective practice*. This constructionist theory is a reaction to the problem solving approach, specifically made to address some of the shortcomings Schön perceived in mainstream design methodology. Schön's starting point is his feeling that the paradigm of technical rationality hampers the training of practitioners in the professions. He believes that the design-component of the professions is underestimated, and that the nature of human design activities is misunderstood. He shows that in the training programmes of professional schools that recognise design as a core activity, design knowledge is defined in terms of generalities about design processes and declarative knowledge needed to solve design problems. No attention is paid to the structure of design problems and the crucial problem of linking process and problem in a concrete design situation. This 'action-oriented', often implicit knowledge cannot be described within the paradigm of technical rationality. But Schön insists that this kind of knowledge is vital for action-oriented professions like design. He does recognise, however, that this implicit 'knowing-in-action' is difficult to describe and convey to students. What can be thought about and taught is the explicit reflection that guides the development of one's knowing-in-action habits. This he calls *reflection-in-action*.

One of the basic assumptions of the theory of technical rationality is that there is a definable design problem to start with. Schön remarks that '

> Although Simon proposes to fill the gap between natural sciences and design practice with a science of design, his science can only be applied to well-formed problems already extracted from situations of practice...'

Schön, on the other hand, does not make any such assumptions about the design problem. The description of design as a reflective conversation concentrates on the structuring role of the designer, setting the task and outlining possible solutions all in one 'framing' action. The strength of this framing action determines the amount of structure in the task.

Schön thus seems to ignore the possible *a priori* structure that design tasks and solutions might have. Schön's failure to link the theories of reflective practice to a model of design tasks means that descriptions of design activities within this paradigm can not benefit from any structure that might be present in the design task. If anywhere, the structure of the design problem should be found in the frame a designer uses. It is a pity that Schön never addressed the questions how frames are made, and what the properties of a good frame would be.

2. Kinds of design problems

These paradigms of design methodology can be the basis for a further exploration of the structure of design problems. A fundamental distinction between kinds of design problems can actually be constructed on the basis of the paradigms of design methodology themselves. This is an initial step towards a typology of design problems.

The key to understanding how the different paradigms view design problems is rooted in their epistemologies. The rational problem solving paradigm is based upon positivistic epistemology, and the paradigm of reflective action is phenomenological in nature. Positivism and phenomenology differ quite strongly in the way subject (the acting person) and object (the outside world) are related. Positivism claims that a person lives in an objective world which can be known through his/her senses; the sensory data is then structured by an internal processing system. This structuring system interprets the data by using basic *a priori* categories. To know the objective world, a person should study it carefully and dispassionately, preferably with scientific methods. In phenomenology the person is not static, but a dynamic, emotive social being with a history and an environment which heavily influences the person's construction of reality. And the subject is influenced (and in the end 'formed') by what he/she perceives. Therefore, person and object are inextricably connected (Merleau Ponty 1992).

Positivism and phenomenology are on opposite sides of the epistemological spectrum (Coyne 1995, Varela 1991). Over the centuries a number of attempts have been made to bridge this gap - but none of these attempts has generally been recognised as successful. However, some relatively recent developments in hermeneutics could be useful when dealing with this dilemma in the specific case of design methodology. Gadamer proposes to bridge this gap by defining the basic operation in the acquisition of knowledge (in this case about design problems) as ***interpretation***. And he claims that this interpretation is a dualistic activity: it is both a *'revealing of what the thing itself already points to'* and *'an attribution of value to something'* (Gadamer 1986) The 'revealing of what the thing itself already points to' could be called *'objective interpretation'*. This is the case when something outside (say, a piece of information) impresses its meaning upon the observer. The 'attribution of value to something' could be called *'subjective interpretation'*. This means that the subject, in an act of will, impresses meaning and value upon something.

If we apply this to design we can see that the type of interpretation that is dominant varies through the phases of design activity, and across design situations.

The decision whether a part of a design activity will involve 'objective' or 'subjective' interpretation ultimately rests with the designer working on the design problem. Empirical evidence has shown (Dorst 1997) that here are a number of influences on this interpretive behaviour of the designer:

• Inasmuch as a design project is a problem solving process for the outside world, it needs to be *controlled* and the design decisions must be justified to the stakeholders. In that case there is an emphasis to objectify the goals and decisions in the design project, to effectively eliminate the implicitness and elements of 'subjective interpretation' from the design activities. Any perception and problem interpretation must then be made explicit and becomes a subject of negotiation between the designer and the stakeholders. Through this process of negotiating, design becomes a more or less 'objective' process, in which problem statements, programmes of requirements, ideas and design concepts are still *made* rather 'subjectively' and implicitly, but in the end are presented explicitly and evaluated in order to settle them and thus make them real objects in the world. 'Objectivity' of the steps in a design process and of the terms used to describe it can thus be considered an artificial *construction* by the designer(s) for special purposes.

• 'Subjective interpretation' can become very important in a design project (or phase) where the design problem is ill-structured. In such a case, subjective structuring is the only way to make sense of the problem. Structuring of the problem can be achieved by imposing personal goals of the designer into the design problem or by subjectively choosing priorities. In design there is no particular pressure to minimise the subjective interpretation of a design problem and its solution: the dominant goal in design is to produce a good design, on cost and on time. A designer thus has the privilege and the problem of working in both an 'objective' and a 'subjective' mode.

• Where a certain design project gives (or demands) freedom of choice to the designer, he/she has to depend on their own interpretation and perception of the problem to produce a result. Then design is essentially a subjective activity, which can be best described in terms of reflective practice. This is particularly true in the conceptual phase of many design processes, but this subjective approach could extend over whole design processes.

• Empirical studies have shown that designers spend quite some time at the beginning of a design assignment to consider what kind of problem they had to deal with. They did this in terms of the constraints of the problem which imposed on their freedom to define their own goals. The freedom depended partly on the assignment that they were given, and partly on their personal style: some designers seemed more at ease with an 'objectivist' approach to design problems, others are more comfortable imposing influential frames on the project right from the beginning.

• Group- or organisational design processes tend to require a large number of objectifying statements and arguments (Valkenburg 2000) to keep everyone on track. This is even more extreme in multidisciplinary

teams, where the basic level of shared understanding necessary for the completion of the job is more difficult to achieve.

As stated before, the decision whether a part of a design activity will involve 'objective' or 'subjective' interpretation ultimately rests with the designer working on the design problem. The design activities in which 'objective interpretation' plays a major role are described well by the rational problem solving paradigm. Activities that involve 'subjective interpretation' are most easily described by the paradigm of reflective practice.

3. The problem of design problems

Somehow, these reflections on the types of underdetermination involved in design and the paradigms of design methodology have not brought us much closer to answering our research questions. We are still far away from developing a basis for the descripton of the structure of design problems and a taxonomy of design problems. If anything, we have been taken away from being able to make a clear taxonomy of design problems by the dual (schizophrenic?) nature of design.

But what is the kind of structure we are looking for? Quite simply, it is the pattern of connections between subproblems. In the discussion above have seen that this pattern is probably not very stable: the Reflective Practice paradigm holds that the problem structure is (1) going to be dependent on the possible solutions, and that (2) it cannot be objectively established. In a later section we will explore these issues further by studying the interaction between problem and solution in a detailed empirical study.

Now we will take another step back, and look at ways in which we could develop a taxonomy of design problems that is not based on an *a priori* 'objective' problem structure. Instead we will look at the (sub)problems designers see or and tackle in their design situation, and develop a method for the empirical study of design as situated problem solving.

3.1 Co-evolution

In a recent paper, Dorst and Cross (2001) have tried to find a way to arrive at a closer description of underdetermined problem solving by using an empirical study to describe the co-evolution of the design problem and the design solution. If we take a closer look at the creation of solutions to underdetermined design problems, it seems to be a much more gradual process, like an evolution. It seems that creative design is not a matter of first fixing the problem (through objective analysis or the imposition of a frame) and then searching for a satisfactory solution concept. Creative design seems more to be a matter of developing and refining together both the formulation of a problem and ideas for a solution, with constant iteration of analysis, synthesis and evaluation processes between the two notional design 'spaces' - problem space and solution space. In creative design, the designer is seeking to generate a matching problem-solution pair, through a 'co-evolution' of the problem and the solution. Our

observations confirm that creative design involves a period of exploration in which problem and solution spaces are evolving and are unstable until (temporarily) fixed by an emergent bridge which identifies a problem-solution pairing. This description of design as the co-evolution of problem and solution leads to the uneasy conclusion that in describing design, we cannot presuppose that there is something like a set 'design problem' at any point in the design process. Can we still theorize about design at all if we let go of the idea that design is a process running from 'a problem' to 'a solution'?

3.2 Design problems as situated problems

The Rational Problem Solving paradigm and the Reflective Practice paradigm have both been developed in the 60's and 70's, largely inspired by developments in AI and the cognitive sciences. The epic endeavour to build intelligent computer systems focussed on the ability of such a system to solve ill-structured problems within an open context, somewhat comparable to designing. The systems were based on a Rational Problem Solving approach, representing the 'relevant aspects' of the world and setting up formal procedures that manipulate these reprentations to solve a problem.   This approach (GOFAI) has failed (Dreyfuss, 1992). Alternative approaches are now developed that are inspired upon the situatedness of problem solving activity (Varela 1991, Winograd 1986, Suchman, 1987). We will now explore whether the considering of design as situated problem solving will help us get closer to a description of what strcuture design problems have.

A fundamental choice that is associated with situated problem solving is that we are first and foremost interested in what design problems are to the designer, seen through the eyes of the designer, in the design situation. This means that we concentrate on the 'local' design problem that a designer faces, and bracket the 'overall' design problem as something of an abstraction (for now). So we will also have to confront the vagueness (i.e. lack of overview) and subjectivity that is involved in local design actions and decisions. Seen from this perspective, 'the design problem' as such does not really exist as an objective entity in the world. There is an amalgamate of different problems that centers around the basic challenge that is described in a design brief. This amalgamate of problems is partly there to be discovered by the designer in the design process, and part of it has to be MADE by the designer. The process of 'approaching a design problem' or 'dealing with a problematic situation' becomes the vital clue to understanding what design problems are. The latter formulation is important: for much of the design project the problem solving steps can be quite logical, routine and implict, without a real choice for the designer. Dreyfuss holds that problematic situations are the results of a 'breakdown' in this normal, fluent problem solving behaviour (the problem becomes 'at hand', in Heidegger's terms). These 'breakdowns' then are the moments of real choice. It thus becomes very important to distinguish and describe the nature of these breakdowns, the critical situations in design (Frankenberger 1996). These breakdowns are the points that Schön describes as 'surprises', the turning points in the designer's reflective conversation with the situation. In the solution of

these breakdowns 'objective' or 'subjective' interpretation can play a role. This is where the existing (but possibly implicit or unknown) structure of the design problem and the structuring actions of the designer meet. A well-structured problem 'leads' the designer (through deduction, or abduction with a clearly dominant result), an ill-structured problem requires something like a framing action.

These fundamental issues show that the study of the structure of design problems is not a straightforward affair at all. And that we just cannot develop a meaningful 'objective' taxonomy of design problems if we can be convinced by Dreyfus and others that it doesn't exist, that there is never a (complete) representation of the design problem in the head of the designer. The only thing now left for us to study is the 'local' network of links that a designer considers while tackling a design problem in the design situation. The subjective nature of this local network of problems means that we need to have a model of how designers approach a problematic situation. In the next section we will look at a model of design expertise that could be the basis for this.

3.3 Levels of expertise

To explore the levels of expertise we now turn to a lecture by Hubert Dreyfus (2003), in which he pointed out that the nature of the problem that is considered in a problem solving situation depends on the level of expertise of the problem solver. Dreyfus distinguishes five distinct levels of expertise, corresponding with five ways of perceiving, interpreting, structuring and solving problems:
(1) A **novice** will consider the objective features of a situation, as they are given by the experts, and will follow strict rules to deal with the problem. For an advanced **beginner** (2) the situational aspects are important, there is a sensitivity to exceptions to the 'hard' rules of the novice. Maxims are used for guidance through the problem situation. A **competent** problem solver (3) works in a radically different way. He selects the elements in a situation that are relevant, and chooses a plan to achieve the goals. This selection and choice can only be made on the basis of a much higher involvement in the design situation than displayed by a novice or an advanced beginner. Problem solving at this level involves the seeking of opportunities, and of building up expectations. There is an emotional attachment, a feeling of responsibility accompanied by a sense of hope, risk, threat, etc. At this level of involvement the problem soolving process takes on a trial-and-error charcater, and there is a clear need for learning and reflection, that was absent in the novice and the beginner. A problem solver that then moves on to be **proficient** (4) immediately sees the most important issues and appropriate plan, and then reasons out what to do. The real **expert** (5) responds to specific situation intuitively, and performs the appropriate action, straightaway. There is no problem solving and reasoning that can be distinguished at this level of working.

Most of these levels are recognizable to anyone involved in teaching design (or in learning to drive a car, for that matter). The definitions of the levels are still sketchy, and not all the steps may be described

unequivocally (this is very much work in progress). The most important step to focus on in this paper is the one from advanced beginner to competent designer that can be recognized in design education too [Dorst 2003]. This is where involvement and reflection come in to change the problem solving process. This is also where there is a radical shift in the perception and interpretation of the problematic situation: we move from a detached view of an 'objective' reality to the involvement and active interpretation of a situation. These fundamentally different ways of looking at problematic situations can actually co-exist in a design project: nobody is an expert on all aspects of design, on some problems we might be novices, at others we might be competent, or experts. Designers display rule-following behaviour, as well as the interpretation and reflection that characterise higher levels of expertise at work. The nature of the design problem as seen by the designer depends on the level of expertise of the designer in solving the problem.

The level of expertise potentially is a central notion in the description of design practice: the choice of paradigm for describing and supporting design processes depends on the level of expertise that the designer has. The rule-following behaviour of the novice and the advanced beginner needs to be described within the framework of the Rational Problem Solving paradigm. The behaviour of the competent designer and higher can be described using both paradigms, with the Reflective Practice paradigm becoming more relevant the closer we are to expert behaviour.

4 Empirical study

To study the structures of design problems as they are perceived or constructed by designers, we need a research method that allows us to describe the behaviour of designers as operations on the design problem. The research method should be neutral, and it should closely follow the design process on a very detailed level. With 'neutral' we mean here that it should be independent of the paradigms, and just provide a 'language'and a way of working that allows us to monitor designing as closely as possible, using the terms the designer him/herself uses. We need to find a way to describe the design behaviour at a such a detailed level that it allows us to see the structure, the patterns of problem-related behaviour emerge, without being completely problem-specific (case-specific).

Other researchers before us have struggled with this problem, and a possible avenue to solve it is to concentrate on the linking behaviour of designers. The patterns of links that designers perceive and develop in the design problem can be traced with 'linkography', a method first used on design by Gabriela Goldschmidt, and later much refined and extended by Remko van der Lugt (van der Lugt 2001). The latter has traced the way designers build upon eachother's ideas within a brainstorming session by constructing a 'link matrix' of such a session. In this method the ideas the designers had were listed on both axes of the matrix, and the connections between ideas could then be noted down as crosses in the cells of the matrix. This simple method already creates a wonderful overview of the way the brainstorming session progressed, and one could clearly see patterns emerge that were particular to the individual brainstorming groups.

For the purposes of our study we do not want to follow a brainstorming session, but a complete design process. And we want to trace the linking behaviour of designers with regard to design problems, design solutions and the link between the two. This means that one matrix will not suffice, but that we need a system of three matrixes: one in which the design subproblems are on both the axes (so we can map the links between the subproblems that a designer makes), one where the elements of the solution are on the two axes, and one in which the subproblems are on one axis, and the solution elements on the other. If we can fill in these matrixes, following the design process step by step, we will be able to trace the perceived problem and the design solution at hand at every moment of the design process. And we can distinguish patterns in the operations the designers perform upon these design problems and solutions. This method has been tested on the data of the 'Delft Protocols Workshop' (Cross 1996), and the first results have shown that the research method does indeed allow us to closely monitor the way designers link parts of the problem and solution.

This research method allows us to view designing in a different way, as it were at a 90 degree angle to the process-focussed methodology: all the design process methods, techniques and strategies described in design methodology will have their influence on the patterns of the matrixes. The two approaches of design problems that are associated with the paradigms of Rational Problem Solving and Reflective Practice will also lead to different patterns - the Rational Problem Solving approach would lead to design behaviour that is initially very much aimed at establishing many links between parts of the design problem. The Reflective Practice approach would be visible from the many jumps between the problem and solution matrixes. Frames can be seen as a complete pattern of problem-solution connections that are imposed upon the existing problem and solution. More experienced designers will have the advantage of having a bigger store of these ready-to-use linking patterns. These will help them structure their design problems and design solutions, and to 'frame' the links between the two.

5. Conclusion

In this paper we have explored ways to describe the structure of design problems. In doing this we have moved away from making an *a priory* taxonomy of design problems, because that inevitably focuses on describing the structures of reasonably determined problems. For design, this will not get us very far: design problems are largely underdetermined. To really capture design, we need to consider the problems as situated problems, as they are seen through the eyes of the designer. Thus the original research question has shifted from the development of a taxonomy of design problems, to a description of critical design situations. This involves the study of the breakdowns that can occur in the flow of design problem solving, and the designer's response to these breakdowns. To describe this response we have turned to the paradigms of design methodology and to the co-evolution model of design processes. We have developed an empirical research method that potentially allows us to monitor this detailed design behaviour.

But these are just descriptions of what happens in a design situation. To really UNDERSTAND why a designer tackles a problematic situation in a certain way we have to turn to a model of design expertise. This is where all the elements we need for a close description of design problem solving behaviour that have been explored in this paper connect. The levels of expertise potentially have the power to coherently describe the ways in which designers perceive, interpret, structure and solve design problems.

The primitive model of design expertise that was presented in this paper needs to be develop further, and validated by empirical research. Without such a model, we cannot really solve the 'Problem of Design Problems'.

References

Alexander, C. (1964) Notes on the synthesis of form, Harvard University press, Cambridge MA.

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

Cross N.G., Christiaans H.H.C.M. and Dorst C.H. (Eds) (1996) Analysing Design Activity, Wiley, Chichester.

Cross, N.G. (ed.) (1984) Developments in Design Methodology, Wiley, Chichester.

Dorst, C.H. (2003) Understanding Design, BIS Publishers, Amsterdam.

Dorst, C.H. and Cross, N.G. (2001) Creativity in the design process: co-evolution of problem-solution, Design Studies 22, page 425-437.

Dorst, C.H. (1997) Describing Design - a Comparison of Paradigms, thesis TUDelft.

Dorst C.H. (1995) Comparing the paradigms of design methodoldogy, In: Hubka, W.(Ed) (1995), Proceedings of ICED 95, Heurista, Zürich.

Dreyfus, H.L. (2003) unpublished notes from the Spinoza lectures, may 26th & june 23rd at the University of Amsterdam.

Dreyfus, H.L. (1992) What Computers still can't do, MIT Press, Cambridge MA.

Frankenberger, E. and Badke-Schaub, P. (1996) Modelling design processes in industry – empirical investigations of design work in practice, In: Akin, O. and Saglamer, G. (Eds) (1996), Proceedings of DMD'96, Istanbul.

Gadamer, H-G. (1986) The relevance of the beautiful and other essays, Cambridge University Press, Cambridge.

van der Lugt, R. (2001) Sketching in idea generation meetings, Thesis TUDelft.

Meijers, A.W.M. (2000) The relational Ontology of Technical Artifacts", In: Kroes, P.K. and Meijers, A.W.M. (2000) The empirical turn in the philosophy of technology, Elsevier Science, Oxford.

van Manen, M. (1990) Researching lived experience, The Althouse Press, Ontario.

Merleau-Ponty M. (1992) Phenomonology of Perception, Routledge, London.

Lawson, B. (1990) How designers think: the design process demystified (2nd ed) Butterworth, London.

Newell, A. and Simon, H.A. (1972) Human Problem Solving, Prentice-Hall, Englewood Cliffs.

Roozenburg, N.F.M. and Eekels, J. (1995) Product Design: Fundamentals and Methods, Wiley, Chichester.

Roozenburg, N.F.M. and Cross, N.G. (1991) "Models of the design process - integrating across the disciplines", In: Hubka, V. Proceedings of ICED 91, Heurista, Zürich.

Schön, D.A. (1983) The Reflective Practitioner, Basic Books, New York.

Simon, H.A. (1992) Sciences of the Artificial, The MIT Press, Cambrigde MA.

Simon, H.A. (1973) The structure of ill-structured problems, Artificial Intelligence, 4, 181-201.

Suchman, L.A. (1987) Plans and situated actions, Cambridge University Press, Cambridge UK.

Valkenburg, A.C. (2000) The Reflective Practice in Product Design Teams, Thesis TU Delft.

Van der Lugt, R. (2001) Sketching in idea generation meetings, thesis TUDelft.

Varela, F.J. Thompson, E. and Rosch, E. (1991) The embodied mind, MIT Press, Cambridge MA.

Winograd, T. and Flores, F. (1986) Understanding computers and cognition, Ablex, Norwood NJ.