

Design Problems and Design Paradoxes

Kees Dorst

Introduction

The problem-solving literature that arose in the 1960s and 1970s in the promising and exciting field of artificial intelligence has had a profound impact on Design Methodology. The introduction of these theories in Design Methodology, at the start of the 1970s, helped to systemize the models and methods of design existing then, and link them to models of problem solving in other fields. There were high hopes that the very nature of design could be captured in a description that was based upon considering design the solution to “ill-structured problems.”

Although there have been many developments since then, the original work on problem solving and the nature of ill-structured problems, written by Herbert Simon, still looms large over the field of design methodology. The rational problem-solving paradigm, based on the conceptual framework that Simon introduced, is still a dominant paradigm in the field.¹ Design models and methods have been developed within this paradigm; the conceptual framework of rational problem solving has become the normal “language” of thinking and talking about design. There also have been many critiques of Simon’s problem-solving approach and its applicability to the field of design, and many of the original statements in the problem-solving theory that deal with design have since been qualified and refined. However, these critiques have not produced a fundamentally different alternative to the conceptual framework.

In this piece, we will revisit the basic assumptions behind Simon’s approach to design, notably the central concept of “ill-structured problem,” and introduce some ideas that could lead to an alternative conceptual framework for thinking about design problems. First, we will revisit the original work by Simon at considerable length, and unearth the assumptions that underlie the conceptual framework that Simon uses to describe ill-structured problems. Then we will deal with some more recent developments within the problem-solving framework, and discuss some of the critiques on the rational problem-solving approach to design—again concentrating on the central notion of ill-structured problems. We will use this critique to propose a fledgling framework of alternative concepts that could be used to augment our understanding of the nature of

1 C.H. Dorst, *Describing Design: A Comparison of Paradigms* (thesis TUDelft, 1997).

“design problems.” We will end with some further reflections and ideas for the development of an alternative framework for describing design.

The Core of Simon’s Theory

In describing the core of Simon’s conceptual framework, we first will concentrate on his classic 1973 paper: “The Structure of Ill-structured Problems.” Unless otherwise stated, all quotes used in this section are taken from that paper. Of course, the conceptual framework that an author uses can be read from any paper, but this paper is especially suited because it is nicely explicit in its application to the field of design—although still broader in its orientation. The most important conclusions have been checked with other (later) papers by Simon.

Within Simon’s theory, the issue of the solution of design problems takes the stage as an example of a wider category of problems—what he terms “ill-structured problems.” In his paper, Simon sets out to explore the relation between ill-structured problems and “well-structured problems.” He starts with the remark that many kinds of problems that often are treated as well-structured probably should be regarded as ill-structured. Even the limited problems (“limited” in the sense of taking place in an enclosed and well-defined world) that are used as standard examples in problem solving and AI literature, such as in playing chess, display elements of ill-structuredness on closer scrutiny:

Even if we regard chess playing as a well-structured problem in the small, by most criteria it must be regarded as an ill-structured problem in the large (over the course of the game).

The stated goal of his paper is to show that there is no real boundary between well-structured problems and ill-structured problems, and therefore no reason to assume that the solution of ill-structured problems would require new and hitherto unknown types of problem-solving processes. To start the comparison, Simon lists the following properties of well-structured problems:

- 1 There is a definite criterion for testing any proposed solution, and a mechanical process for applying the criterion.
- 2 There is at least one problem space in which it can be represented as the initial problem state, the goal state, and all other states that may be reached, or considered, in the course of attempting a solution to the problem.
- 3 Attainable state changes (legal moves) can be represented in a problem space, as transitions from given states to the states directly attainable from them. But considerable moves, whether legal or not, also can be represented—that is, all transitions from one considerable state to another.

- 4 Any knowledge that the problem solver can acquire about the problem can be represented in one or more problem spaces.
- 5 If the actual problem involves acting upon the external world, then the definition of state-changes and of the effect upon the state of applying any operator reflect with complete accuracy in one or more problem spaces the laws (laws of nature) that govern the external world.
- 6 All of these conditions hold in the strong sense that the basic processes postulated require only practicable amounts of computation, and the information postulated is effectively available to the processes—i.e., available with the help of only practicable amounts of search.

In the cases where all these rules apply, they allow such a well-structured problem to be solved by a “general problem solver,” a computer program that follows preset rules to arrive at a solution. Simon later adds an extra criterion that further limits the actions that are allowed within the problem-solving process:

What some notions of well-structuredness require, however, is that these capabilities be defined in advance, and that we do not allow the problem solver to introduce new resources that “occur” to him in the course of his solution efforts. If this condition is imposed, a problem that admits restructuring through the introduction of such new resources would be an ill-structured problem. A problem that is not solvable with reasonable amount of computation when all knowledge must be expressed in terms of the original problem space may be easily solvable if the problem solver is allowed to use knowledge in another space.

Two important points can be picked up from these definitions. Apparently, if the problem-solving effort involves learning, or the redefinition of the problem, the problem cannot be considered well-structured. And there is a methodological point to be made: apparently, if we take item six in the definition of well-structured problems seriously, the ill-structuredness of a problem depends on the solution methods that are available to solve it. This opens up the way for suspecting that the ill-structuredness of a problem may not be an *a priori* property of the problem itself, but is linked to the capabilities of the problem solver. In this way, the subject that does the problem solving actually influences the very nature of the problem. Simon goes on to explain that:

In general, the problems presented to problem solvers by the world are best regarded as ill-structured problems.

Nevertheless, Simon maintains that the problem-solving theory that is based upon the solution of well-structured problems should serve as the basis for **all** problem solving. He has been criticized for this standpoint, and later concedes that:

... there is merit to the claim that much problem-solving effort is directed at structuring problems, and only a fraction of it at solving problems once they are structured.

There is a basic assumption here that even though well-structured problems as such do not exist in the real world, the construction of well-structured problems from ill-structured problems is the way to solve an ill-structured problem. Simon then illustrates the solving of ill-structured problems by taking an example from design. The example involves designing a house, and concentrates on the technical problem of designing the layout of the house. In this case, the structuring actions that turn the ill-structured design problem into a well-structured problem are done by the architect:

Additional specification will be obtained from the dialogue between architect and client, but the totality of that dialogue will still leave the design goals quite incompletely specified. The more distinguished the architect, the less expectation that the client should provide the constraints.

This is quite plausible, but Simon ignores the other conclusion one could draw from this example: namely, that even in the case of such a technical problem, with clear variables that allow for technical reasoning, and with the involvement of only one stakeholder, even here subjectivity creeps into the problem-solving process by the actions needed to construct a solvable problem. This means that, for the problem-solving theory to hold up as a good basis for the description of design, we now also need a detailed description of the problem solver, including an account of the earlier knowledge that the problem solver potentially brings to bear on this situation. One could even conclude that an ill-structured problem can't be modeled without taking these properties of the problem solver into account. The interpretation of the problem is important, even in the simple example that Simon describes. Interpretation becomes even more important when we see that design is a process of multiple steps, not a one-off decision making situation. New interpretations will be based upon the interpretation that has been taking place in the earlier steps of the problem-solving process:

As a matter of fact, the whole procedure could be organized as a system of productions, in which the elements already invoked from memory and the aspects of the design situation already arrived at up to any given point, would serve as the stimuli to evoke the next set of elements.

This means that in a multistep problem-solving process, each problem solver will get the chance to pile interpretation upon interpretation, and thus end up taking the problem-solving processes in completely different directions. Therefore, the use of memory and subjective interpretation becomes a major influence on the problem-solving behavior of designers. If we take this seriously, then it undermines the very idea of having one, knowable problem at the start of the problem-solving process. But Simon misses this point:

... the architect will find himself working on a problem which, perhaps beginning in an ill-structured state, soon converts itself through evocation from memory into a well-structured problem.

This statement is not supported by the data provided by Simon in his own example. It is, in fact, a restatement of the assumption that only well-structured problems can be solved. This is one of the major points where the applicability of Simon's problem-solving theory to design has been questioned. It is important to note that, even if we were to agree with Simon that design problem solving would be based on the "normal" solution of well-structured problems, then this step of conversion becomes a major part of the problem-solving activity. This should be specified for the problem-solving theory to be complete. Simon later partially agreed to this by introducing an unspecified "noticing- and-evoking mechanism," speaking about the need for an "indexed memory," and placing the design process in an "effective problem space":

... the effective problem space will undergo continuing change throughout the course of the game (author's note: *problem-solving activity*), moving from one subspace to another of the large space defined by the contents of the long-term memory.

Here again, the course of the problem-solving process and the very structure of the ill-structured problem are determined by the possibilities for action that the problem solver considers. These possibilities for action are closely linked to the interpretation of the problem and the content of the acting subject's memory.

Simon concludes his paper:

... the boundary between well-structured and ill-structured problem solving is indeed a vague and fluid boundary. There appears to be no reason to suppose that concepts as yet uninvented and unknown stand between us and the fuller exploration of those domains that are most obviously and visibly ill-structured. It suggests that there may be nothing other than the size of the knowledge base to distinguish ill-structured problems from well-structured problems, and that general problem-solving mechanisms

that have shown themselves to be efficacious for handling large, albeit apparently well-structured domains should be extendable to ill-structured domains without any need for introducing qualitatively new components.

This conclusion is not supported by the data, and contains a logical weakness: the fact that it is hard to draw a line that distinguishes between well-structured problems and ill-structured problems doesn't mean that there is no difference. There may be elements within the process of solving ill-structured problems that can actually be more or less straightforward steps (that can be considered well-structured problems), but that doesn't mean that the solving of ill-structured problems can be reduced to these straightforward steps. There is no evidence to support the claim that both kinds of problem solving are the same. The problem here is that Simon models well-structured problems and ill-structured problems in the same way. He never escapes from the circularity in his argument, and shoves aside the obvious differences between well-structured problems and ill-structured problems.

Later Developments

The rational problem-solving paradigm has become a powerful tool for the modeling of design, inspiring and permeating a large part of design methodology. However, the fundamental weaknesses in the conceptual framework that were unearthed in the last section also can be recognized in these later developments of the problem-solving approach to designing. The main thesis in this paper will be that these weaknesses are such an integral part of the problem-solving inheritance that they cannot easily be solved from within the rational problem-solving paradigm. Two examples might illustrate this point.

In the substantial body of work on the "Function-Behavior-Structure" model of design that has been developed by the research group at the Key Centre for Design Computing and Cognition, under the supervision of John Gero, we can find several echoes of the same difficulties. For instance, the FBS model ascribes an equally large role to the use of "design prototypes" in determining the "framing" of the design problem, as Simon does to the "memory" and "experience" that a problem solver needs to transform an ill-structured problem into a well-structured one. This large role for "experience" and "prototypes" leads to grave methodological difficulties. Because of the very open-ended way in which the use of "design prototypes" is described in the FBS model, they potentially make up a vital part of the design process, actually bypassing the design process that is modeled in the core FBS model.² The neat and clear design process model loses most of its value if it is preceded by a very messy and overwhelmingly influential step called "the adoption of a prototype."

2 C.H. Dorst and P.E. Vermaas, "John Gero's Function-Behavior-Structure Model of Designing: A Critical Analysis" *Research in Engineering Design* 16 (2005): 17–26.

In a recent paper, Dorst and Cross³ have tried to find a way to arrive at a closer description of problem solving of ill-structured problems by using an empirical study to analyze and describe the design process as a “coevolution” of the design problem and the design solution. This coevolution model of design is based on the work by Maher et al.⁴

Based on their empirical study, they observe that the creation of solutions to ill-structured design problems seems to be a very gradual process—an evolution. Their analysis shows 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 coevolution of the problem and the solution. 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. The description of design as the coevolution 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.

This leads to some very pertinent methodological questions. Can we still describe design in terms of problem-solving theories if we have to abandon the idea that the “design problem” can be identified at all? What then is the meaning of saying that design is a process running from “a problem” to “a solution”? We can probably stick to the problem-solving theory of design only if we abandon the idea that there is a definable problem at the start of the design process, and postulate that it will be constructed later on. This then begs the question how this problem is constructed, and whether this process of “problem construction” can be modeled at all. Also, if this process of problem construction could be modeled, whether that modeling should be done within the rational problem-solving paradigm, or outside of it.

In the next sections, we will introduce two different approaches that already have been taken to tackle this problem. First, we will consider the work of Dreyfus and Suchman, modeling design problems as situated problems; and then we will look at Hatchuel’s ideas on “extended rationality.” Finally, we will use this critique to present an idea for a fledgling model of design problems based on a radically different set of concepts.

3 C.H. Dorst and N.G. Cross, “Creativity in the Design Process: Co-evolution of Problem-solution,” *Design Studies* 22 (2001): 425–37.

4 M.L. Maher, J. Poon, and S. Boulanger, “Formalizing Design Exploration as Co-evolution: A Combined Gene Approach” in *Advances in Formal Design Methods for CAD*, J.S. Gero and F. Sudweeks, eds. (London: Chapman and Hall, 1996).

New Approaches

The “rational problem-solving paradigm” developed in the 1960s and ‘70s largely was inspired by developments in AI and the cognitive sciences. The epic endeavor to build intelligent computer systems focused on the ability of such a system to solve ill-structured problems within an open context—somewhat comparable to designing. These systems, based on a rational problem-solving approach, represented the “relevant aspects” of the world, and set up formal procedures to manipulate these representations in order to solve a problem. This approach has failed.⁵ Alternative approaches were developed based on situating problem-solving activity.^{6,7,8} We will now explore whether considering design as situated problem solving will help us get closer to developing an alternative description of design problem solving.

Situated Problem Solving

The description of design as a situated activity involves two important shifts in standpoint. The first consideration in situated problem solving is the design problem as 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 ignore the “overall” design problem as something of an abstraction. We also have to deal with the vagueness (i.e., lack of overview) and subjectivity inherent 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. It is an amalgamation of different problems centered on the basic challenge described in a design brief. This amalgamation of problems discovered by the designer in the design process is partially created by the designer. The process of “approaching a design problem” or “dealing with a problematic situation” is a vital clue to understanding what design problems are. The second fundamental shift in standpoint is that, for much of the design project, the problem-solving steps can be quite logical, routine, and implicit; without any real choice by the designer. Dreyfus holds that problematic situations are the result of a “breakdown” in this normal, fluent problem-solving behavior. (The problem becomes “at hand,” in Heidegger’s terms.) These “breakdowns” are then the moments of real choice. It thus becomes very important to distinguish and describe the nature of these breakdowns—the critical situations in design.⁹ These breakdowns are the points that Schön, in his work on reflective practice, describes as “surprises.”¹⁰ Schön describes them as the turning points in the designer’s reflective conversation with the situation. Please note that the definition of a “design problem” has been narrowed, and limited to the situations where routine problem solving has failed.

If we can be convinced by Dreyfus and others that there is never a (complete) representation of the design problem in the head of the designer, then the only thing left for us to study is the “local”

-
- 5 H.L. Dreyfus, “Intelligence without Representation: Merleau-Ponty’s Critique of Mental Representation,” *Phenomenology and the Cognitive Sciences* 1 (2002): 367–383.
 - 6 F.J. Varela, E. Thompson, and E. Rosch, *The Embodied Mind* (Cambridge, MA: MIT Press, 1991).
 - 7 T. Winograd and F. Flores, *Understanding Computers and Cognition* (Norwood, NJ: Ablex Publishing, 1986).
 - 8 L.A. Suchman, *Plans and Situated Actions* (Cambridge: Cambridge University Press, 1987).
 - 9 E. Frankenberger and P. Badke-Schaub, “Modeling Design Processes in Industry: Empirical Investigations of Design Work in Practice” in *Proceedings of DMD’96*, O. Akin et al., eds. (Istanbul, 1996).
 - 10 D.A. Schön, *The Reflective Practitioner* (New York: Basic Books, 1983).

network of links that a designer considers while tackling a design problem in the design situation. The incompleteness and subjective nature of this local network of problems means that we need a model of how designers approach a problematic situation.

Problem Solving and Design

Hatchuel¹¹ analyzes the work of Simon on design in its original context, as part of Simon's bigger project on the development of a theory about "bounded rationality." The aim of this project was to "explain human behavior by simple and constrained, yet informed, decision rules." The bounded-rationality project spans Simon's work in economics, artificial intelligence, and design. This lifelong background project can help us understand what Simon is trying to achieve in his writings on design. It also explains the strong emphasis that Simon placed on the solid ground provided by well-structured problems, and the rules for solving them, that we have seen in the section describing the core of Simon's theory.

For Simon, creativity and discovery in science, art, and design were all potentially describable within the bounded rationality perspective. In developing this perspective, Simon sees a strong theory of design as crucial. The possibility to develop a strong theory on design (the archetypal "science of the artificial") within this general framework of bounded rationality serves as a litmus test for the bigger project itself. Simon's drive to achieve this leads to poetic statements:

The proper study of mankind is said to be man ... If I have made my case, then we can conclude that, in larger part, the proper study of mankind is the science of design, not only as the professional component of a technical education but as a core discipline for every liberally educated person.¹²

Hatchuel argues that Simon is overeager in his efforts to incorporate design within the general bounded-rationality problem-solving theory. Hatchuel illustrates the distinction that he thinks needs to be made between design and problem solving by an example in which two problem situations are compared. He pictures a group of friends coming together on a Saturday night. The one problem situation is that they are "looking for a good movie in town"; the other problem situation is that they set out to "have a party." The first situation is considered to be "problem solving," while the second situation is, in Hatchuel's terms, a real design project. Hatchuel argues that there are three important differences between these situations:

- A. The first difference is that the design situation includes the (unexpected) expansion of the initial concepts in which the situation is initially framed ("a party"). This makes the solution process a "project" instead of a "problem." There is no dominant design for what a party should be, so imagination needs to be applied at this very fundamental level.

11 A. Hatchuel, "Towards Design Theory and Expandable Rationality: The Unfinished Program of Herbert Simon," *Journal of Management and Governance* 5:3-4 (2002).

12 H.A. Simon, *Sciences of the Artificial* (Cambridge, MA: The MIT Press, 1992), 159.

- B. A second difference is that the design situation requires the design and use of “learning devices” in order to get to a solution. These “learning devices” are sub-processes that help us “learn about what has to be learned or should be learned.” They include experiments and simulation techniques.
- C. Finally, in designing, the understanding and designing of the social interactions is part of the design process itself. The group of friends needs to develop a way of reaching a solution that cannot be supposed to exist before the design situation arises. This point comes very close to the work of Louis Bucciarelli, who claims that: “Design is *fundamentally* a social process.”¹³

From this comparison, we can conclude that design undoubtedly includes stretches of ill-structured problem solving, but that it also contains other processes. For Hatchuel, design includes problem solving, but it cannot be reduced to problem solving. He states that any model or description method that tries to reduce design to problem solving is bound to miss important aspects of the design activity. This observation ties together our earlier conclusions, and the remarks made on the modeling of design as coevolution and situating design problem-solving activities.

All of this means that the very notion of “design problem” becomes extremely problematic. If the “design problem” in general is not knowable at any specific point in the design process;¹⁴ and if it is evolving in the design process—at least until the creation of the design concept, and possibly beyond that point;¹⁵ and if the connotations of the very concepts that are used to describe a “design problem” are shifting as a part of the design effort;¹⁶ then we need to radically reconsider our use of the term “design problem.” The fundamental question that now presents itself is: What is the real meaning the term “design problem,” and how we can use it in design methodology?

To explore this, we will use a philosophical technique called “bracketing” that was pioneered by the phenomenologist Husserl in the early years of the twentieth century. In bracketing, we first establish that the notion of design problem, though deceptively simple, is just too complex and complicated to be useful in studying design. We have tried to show in this paper that the notion of design problem is so riddled with difficulties that it actually is obscuring our vision of the phenomenon that it tries to cover. So we propose to “bracket” the notion of design problem, meaning that we are temporarily going to describe the underlying phenomenon without using the term itself. Once we have this description and analysis of the underlying phenomenon, the bracketed word can be reintro-

13 L.L. Bucciarelli, *Designing Engineers* (Cambridge, MA: MIT Press, 1994).

14 H.L. Dreyfus, “Intelligence without Representation— Merleau-Ponty’s Critique of Mental Representation,” *Phenomenology and the Cognitive Sciences* 1 (2002): 367–383.

15 C.H. Dorst and N.G. Cross, “Creativity in the Design Process: Co-evolution of Problem-solution,” *Design Studies* 22 (2001): 425–37.

16 A. Hatchuel, “Towards Design Theory and Expandable Rationality: The Unfinished Program of Herbert Simon,” *Journal of Management and Governance* 5:3–4 (2002).

duced, in a way that it is better connected to the other terms that are used to describe the phenomenon within the nomological network. This latter step is important: if the bracketed word is widely used in vernacular descriptions of the subject (as surely is the case for “design problem”), then it would be foolish to ignore that use.

Paradoxes and the Modeling of Design as a Discursive Activity

In this section, we will attempt to create a new description of the design situation without using “design problem,” while accommodating some of the difficulties that we have encountered in our analysis of the use of the term within the rational problem-solving paradigm. Setting up of an alternative conceptual framework is an open-ended problem in itself: design can be described in numerous ways. For reasons that go beyond the scope of this paper, we have chosen to develop a fledgling theory of design centered on “paradox” and “discourse.” The reader should take the description of design that is presented in this subsection as one example of many possible ways to describe the same phenomenon. We hope to inspire the reader to develop additional ways.

The use of the term “paradox” is inspired by the work of Caroline Whitbeck. In her book *Ethics in Engineering Practice and Research*, she remarks:

... The initial assumption [author’s note: within moral philosophy] that a conflict is irresolvable is misguided, because it defeats any attempt to do what design engineers often do so well, namely, to satisfy potentially conflicting considerations simultaneously.¹⁷

This description of paradoxical situations defines the nature of the problematic relationship that designers and engineers are dealing with through their design thinking.¹⁸ “Paradox” is used here in the sense of a complex statement that consists of two or more conflicting statements. In the initial state of the paradoxical problem situation, all the statements that make up the paradox are true or valid, but they cannot be combined. A paradox, a real opposition of views, standpoints, or requirements, thus requires a redefinition of the problematic situation in order to create a solution. An example from product design would be that a certain product, that cannot be moved, needs to be there to perform its function at one moment in time, and it needs to be invisible and not take up space at another moment in time. The creation of solutions to a paradoxical design situation often requires the development and creative redefinition of that situation.

The elementary statements that make up the paradox, and the viewpoints and ways of thinking that underlie these statements, will now be described in terms of “discourses.” The term “discourse” was introduced by the philosopher Michel Foucault, most extensively in his book *The Archaeology of Knowledge*.¹⁹ He uses

17 C. Whitbeck, *Ethics in Engineering Practice and Research* (Cambridge: Cambridge University Press, 1998).

18 This situation of being hampered in the normal, routine problem-solving activity has been described in terms of “surprises” (Schön, 1983), or “critical situations” (Frankenberger, 1996).

19 M. Foucault, *The Archeology of Knowledge* (London: Routledge, 1969, 1989).

this term to describe the complete structure of terms and relationships that lie at the basis of the thinking and discussions within an area of human activity. These terms and relationships can, for instance, be captured in textbooks and in well-known examples of “the normal way of working.” Because the terms and relationships within a discourse make up the very elements of human thought, the discourse in a field spans the *complete* breadth of human thinking within that domain. “Discourse” can be loosely compared to Kuhn’s use of “paradigms,”²⁰ but Foucault doesn’t support Kuhn’s idea of the wholesale “revolutions” as the origin of these discourses. In his own work, Foucault traces the changes in the meaning and use of key concepts within a domain, rather than looking for revolutions.

In most design disciplines, there are many discourses that somehow have to be linked in the creation of a design solution. In product design practice, for example, relevant discourses include the bodies of thought about technology, form an aesthetics, ergonomics, etc. These are called the “aspects” of a design.²¹ Discourses also can be embodied in a design situation by the roles and the value systems of the different stakeholders involved in the project. The creation of a solution to the paradoxical design situation thus also becomes a social process.

The designer, in his/her paradoxical problematic situation, needs to construct a design that transcends or connects the different discourses, in a general sense (by the construction of a meta-discourse), or just in the concrete instance of the design-to-be-developed. To do this, the designer has to step out of the ways of thinking embodied in the discourses. This step is likely to include a strong intuitive element. Based upon a clear understanding of the discourses, and upon earlier experiences with paradoxical situations, a solution is created that needs to be evaluated from the standpoints of all the different discourses (i.e., to see that the solution is valuable within the relevant discourses). Designers use their understanding of the ways of thinking within the different discourses to create a framework in which a solution is possible for the paradoxical situation. The paradoxical problem situation works as both a trigger to creative imagination and as a context for the evaluation of the design. For the solution to *be* a solution, it needs to be recognized as such in the contexts of all the relevant discourses. (In practice, this often means, first and foremost, that it should be acceptable to all the relevant stakeholders.)

Discussion

In this last section, we have constructed a model of design in which the nature of “design problems” is further specified. A “design problem” is taken as a paradox, made up out of the clash of conflicting discourses. The nature of creative design is the forging of connections between these discourses, on a general level or in the concrete design. It should be stressed that this is just a fledgling theory, meant

20 T.S. Kuhn, *The Structure of Scientific Revolutions* (Chicago: University of Chicago Press, 1962, 1969).

21 C.H. Dorst, *Describing Design: A Comparison of Paradigms* (thesis TUDelft, 1997).

to solve some methodological problems, but undoubtedly creating new ones. At least it does shed some new light upon the three main problems with the use of the term “design problems” in design methodology identified earlier:

1. *The “design problem” is not knowable at any specific point in the design process.*²² With the adoption of the design situation as the unit of description, the question of defining the design problem as a whole becomes irrelevant. The paradox that drives the design process within a problematic situation, at a certain moment in the design process, should be determined from the designers’ actions and words. The next task we then encounter in the quest to really understand design is, of course, to define the structure of the discourses. This could be difficult, although Foucault has developed some basic methodologies for this in his original work on the history of the discourse on mental illness.
2. *The “design problem” is hard to identify because it evolves in the design process.*²³ This is partly covered above. We could add that the discourses hardly evolve within a design project, but that paradoxes (the point at which the discourses clash, and the way in which they do) may evolve throughout the design project.
3. *The connotations of the very concepts that are used to describe a “design problem” are shifting as a part of the design effort.*²⁴ The central notions that make up the paradoxes the designers are dealing with indeed are meant to shift in the course of creating a solution. A clear view of the original discourses that play a part in the design project will provide an anchoring point for understanding these shifts.

This model needs to be extended much further, for instance by defining the link between the notion of discourses and the aspects of a design, the stakeholders involved in a design project, and the designer’s level of expertise. But we must leave this here for now.

Concluding Remarks

We hope to have effectively argued that the conceptual framework that underlies much of design methodology, while perhaps not flawed, is full of assumptions that may be questioned. Some of the problems inherent in Simon’s theory are inherited by people using the conceptual framework that he introduced. Our argument has focused on the term “design problem.” The widespread use of this term in the vernacular discussions within and about design make it one of the basic terms in a methodological description of design activities. But we hope to have demonstrated that the term “design problem” is very problematic in a scientific context. In the scientific

22 H.L. Dreyfus, “Intelligence without Representation— Merleau-Ponty’s Critique of Mental Representation,” *Phenomenology and the Cognitive Sciences* 1 (2002): 367–383.

23 C.H. Dorst and N.G. Cross, “Creativity in the Design Process: Co-evolution of Problem-solution,” *Design Studies* 22 (2001): 425–37.

24 A. Hatchuel, “Towards Design Theory and Expandable Rationality: The Unfinished Program of Herbert Simon,” *Journal of Management and Governance* 5:3–4 (2002).

study of design, we cannot say that the design activity consists of reasoning from “a design problem” to “a solution”—at least it becomes meaningless to say so if we cannot define the notion of “design problem” or pin it down in empirical descriptions of design activity.

Temporarily bracketing the term “design problem” allows new frames of reference and descriptions of the design activity to emerge. Within this paper, that process has resulted in an alternative way to describe the design as *the resolution of paradoxes between discourses in a design situation*. This alternative way of describing design potentially sheds new light on the nature of design, and on the kind of creativity that is part and parcel of design. The next step would be to confront this new description of design with the existing models and methods within design methodology, and see if this produces interesting insights into the nature of design.

Additional References

H.L. Dreyfus, *What Computers Still Can't Do* (Cambridge, MA: MIT Press, 2002).

J.S. Gero, K.W. Tham, and H.S. Lee, “Behavior: A Link between Function and Structure in Design” in *Intelligent Computer-Aided Design*, D.C. Brown et al., eds. (Amsterdam: Elsevier, 1992), 193–225.

H.A. Simon, “The Structure of Ill-structured Problems,” *Artificial Intelligence* 4 (1973): 181–201.

This article has been cited by:

1. Colette Andreucci, Marjolaine Chatoney, Jacques Ginestie. 2010. The systemic approach to technological education: effects of transferred learning in resolving a physics problem. *International Journal of Technology and Design Education* . [[CrossRef](#)]
2. Stella Tan, Gavin Melles, Nicolette Lee. 2009. Graphic designers' activities during the conceptual design phase of client-initiated projects. *Art, Design & Communication in Higher Education* **8**:1, 85-92. [[CrossRef](#)]
3. Ho Cheong Lee, Ming Xi Tang. 2009. Evolving product form designs using parametric shape grammars integrated with genetic programming. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **23**:02, 131. [[CrossRef](#)]
4. Natalie Ebenreuter. 2007. The dynamics of design. *Kybernetes* **36**:9/10, 1318-1328. [[CrossRef](#)]