



0199

Both Generic Design and Different Forms of Designing

W. J. Visser

INRIA, Rocquencourt, France | willemien.visser@inria.fr

1. Introduction

"Activities as diverse as software design, architectural design, naming and letter-writing appear to have much in common" is, based on a series of studies on these diverse activities, one of Thomas and Carroll (1979/1984)'s conclusions. "Designers have strong intuitions to the effect that there are important differences between the cognitive processes involved in design problem solving tasks, such as architecture or engineering, and nondesign problem solving tasks, such as playing chess or doing medical diagnosis." (Goel, 1994, p. 53) With this sentence, Vinod Goel opens his paper "Comparison of design and nondesign problem spaces" (1994), a publication that, together with an older paper co-authored by Pirolli (1989), is often referred to by researchers from various domains of design. According to Goel, the generic-design principle "was one of the corner stone premises of the design methodology movement" (1994, p. 53). The author refers to authors such as Alexander (1964), Archer (1965/1984), Jones (1963/1984), and, of course, Cross (1986; 1984). The design methodology movement was not based on an analysis of the cognitive activities underlying design. In their papers, Goel and Pirolli aim to "motivate the notion of generic design within information-processing theory" (Goel & Pirolli, 1989), that is, within the framework developed by Newell and Simon (1972) in order to analyze problem-solving activities from a cognitive viewpoint. Goel and Pirolli indeed "differentiate design problem solving from nondesign problem solving by identifying major invariants in the design problem space" (Goel & Pirolli, 1989).

Nowadays, using Google with the keyword "generic design" leads to research and applications in the domains of AI and knowledge-acquisition (e.g. KADS and successor work), based on Chandrasekaran (1983)'s "generic tasks", for example, and on generic design methods, and other "generic design agents". These are thus all normatively based approaches to design, whereas Goel and Pirolli's analysis was concerned with, and based on a cognitively oriented descriptive analysis of design and nondesign activities.

In the domain of cognitive design research, even if it is often not stated explicitly, the tendency is to aim models that generalize across design tasks in different domains. In spite of this more or less implicit adherence



to the generic-design hypothesis, and the frequent quotation of Goel and Pirolli's work, there has been no further research to corroborate the hypothesis, however.

This paper aims to take up again the discussion of the generic-design idea. It defends, however, a second hypothesis: even if we adopt a generic-design stance, we claim that designing also takes different forms depending on the nature of the artifact. In the main section of this paper, we propose a series of candidates for dimensions that allow a distinction to be made between these different forms of design.

2. Generic Design, But *Also* Different Forms of Design

Besides important similarities between design activities in different domains, there are also important differences between design tasks and nondesign tasks. This double observation is the basis of the generic-design hypothesis. Designing is a cognitive activity whose implementation (1) in design tasks in different application domains has commonalities, and (2) has distinctive characteristics from other cognitive activities. Except for Goel and Pirolli's work (Goel, 1994; Goel & Pirolli, 1989), however, the idea has not been substantiated through comparative cognitive analyses.

We adhere to the generic-design hypothesis. Yet, we suppose that the nature of the artifact under design introduces specificities in the corresponding design activities, and therefore leads to different "forms" of design. We thus advance the hypothesis that, in spite of the validity of the generic-design idea, designing may take different forms, depending on dimensions of the artifact that is to be designed. Our hypothesis takes the following form:

- (1) design thinking has distinctive characteristics from other cognitive activities;
- (2) there are commonalities between the implementations of design thinking in design tasks in different application domains;
- (3) there are also differences between these implementations of design thinking in different domains;
- (4) but these differences do not reinstate commonalities between designing and other cognitive activities, whereas the commonalities between all the different forms of design thinking are sufficiently distinctive from the characteristics of other cognitive activities, to consider design a specific cognitive activity.

In this paper, we will only be concerned with the third point, and introduce material that supports it. The first and the second points have been discussed in detail by Goel and Pirolli (Goel, 1994; Goel & Pirolli, 1989). The fourth point remains completely hypothetical and requires new empirical research comparable to that conducted by Goel and Pirolli, but in professional, that is "real" design situations (Goel and Pirolli conducted their work in artificially restricted laboratory conditions).



3. Different Forms of Design

The idea that there may be different forms of design has only been hinted at in informal discussions and without any presentation of empirical or theoretical evidence (Löwgren, 1995; Ullman, Dieterich, & Stauffer, 1988). The engineering-design methodologists Hubka and Eder (1987), for example, assert that "the object of a design activity, what is being designed,... substantially influences the design process" (p. 124).

This paper will propose some potential dimensions underlying differences between forms of design (an earlier presentation of these dimensions has been made in Author, 2006). Our discussion will be mainly allusive, introducing material that needs to be further analyzed and developed. We present a number of possible directions, for other researchers to follow—or modify—and complete them.

We do not deal with the effect that the use of different design methodologies may have on designing and on the resulting design (as shown, e.g., by Lee & Pennington, 1994, for software design using an object-oriented or a procedural paradigm).

3.1 Maturity of a domain

This first dimension is based on the idea that tasks that are rather well circumscribed may provide less difficulties for those who are to perform them than tasks whose definition is quite open. One may suppose that this task characteristic varies with the maturity of a domain. This supposition finds some support in the NSF "Science of Design" program (Science of Design, 2004), which aims to "develop a set of scientific [design] principles to guide the design of software-intensive systems". An underlying idea of this program is that "in fields more mature than computer science [such as architecture and other engineering disciplines, such as civil or chemical engineering], design methodology has traditionally relied heavily on constructs such as languages and notational conventions, modularity principles, composition rules, methodical decision procedures and handbooks of codified experience.... However, the design of software-intensive systems is more often done using rough guidelines, intuition and experiential knowledge." Cognitive design research has frequently shown the difficulty of designers' effectively working according to design methodology prescriptions (Carroll & Rosson, 1985; Author, 2003; Author & Colleague, 1990). One may nevertheless suppose that being familiar with the constructs and other "tools" that have been developed for the tasks in a particular domain may be of influence on—facilitate, in this case—designers' activity.

One may notice that related to the idea that underlies the present dimension and that is only touched upon here, is the entire discussion of well-defined vs. ill-defined problems and the implications for the nature of the activities that are concerned with these problems. Here, we can only mention this question (see Author, 2006).

3.2 Use of internal versus external representations

In recent years, we have developed the idea that designing is most appropriately qualified as the construction of representations (Author, 2006). Different types and forms of representations are thus the central structures



that are worked on in the design activity. Differences between artifacts that depend on, or are related to representational issues are thus prone to lead to different forms of design.

A first difference is that between the role and use of internal and external representations. On the basis of research on the processing of these representations, Zhang & Norman (1994) claim that the two different types of representations preferentially activate different types of processes, that is, cognitive (activated by internal representations) versus perceptive (activated by external representations) (p. 118). With Scaife and Rogers (1996), we suppose that things are less systematic, and more complex. Nevertheless, one may expect that the use of internal and that of external representations involves processing differences. As a consequence, designing may differ between domains depending on the importance that external representations play, or may play, in these domains. One may thus suppose that, for example, architectural or mechanical design differ from design of procedures or organizations. This point is particularized in the following subsection.

3.3 Use of graphical representations versus other types of external representations

This dimension is a specification of the previous, rather general dimension. One of the factors underlying the differences that are supposed to exist between software and other types of design (and that will only be alluded to here, see Author, 2006 for a more detailed discussion) may be the different types of external representations primarily used in these domains. The possibilities that are offered by graphical representations compared to other types of external representations (especially, ease of visualization and manipulation, and their corollaries) facilitate, for example, simulation and other forms of evaluation.

This observation probably does not only apply to software design in its classical, that is non-visual forms. It may also hold for the design of other symbolic artifacts, such as other procedures, plans, and organizational structures.

3.4 Dependencies between function and form

This distinction opposes design in domains where function and form can be aligned, to design in domains where individual forms may perform many functions. In the first type of domain, to each particular form corresponds a particular function. In software design for example, to a functional decomposition corresponds more or less directly a structural decomposition. In the second type of domain, for example mechanical design, each design decision can affect each subsequent decision, because a goal may be achieved, structurally, by modifying a previously specified form rather than by introducing a new one (Ullman et al., 1988). These differences may have consequences for designers' decomposition activities.

3.5 "Designing in space versus time" (Thomas & Carroll, 1979/1984)

Research comparing problems governed by temporal or spatial constraints has shown that designers deal differently with these constraints (Colleague, Author, & Colleague, 2004; detailed in Author, 2004) (see also Thomas & Carroll, 1979/1984). However, research has not yet settled clearly the specific relative ease and difficulty involved in the different types of design that preferentially implement these two types of constraints



(for temporal constraints, especially different types of planning). It has even less identified the underlying factors.

3.6 Design of structures versus design of processes

This dimension is related to the previous one. Structures (states) are not necessarily spatially constrained, but processes have necessarily temporal characteristics. By analogy to the differences between the cognitive treatment of spatial and temporal constraints, one may expect that structures and processes are represented differently (especially mentally, but also externally), thus processed differently, and therefore leading to different design activities (cf. Clancey, 1985's distinction between configuration and planning).

3.7 Distance between design concept and final product

Löwgren (1995, p. 94) opposes "external" software design ("design of the external behavior and appearance of the product, the services it offers to users and its place in the organization"; cf. "interactive software" as opposed to "embedded software") to other types of design, for example, architectural and engineering design. In external software design "it is technically possible to evolve a software prototype into a final product"—something difficult (or even impossible) in these other fields. Therefore, in domains such as external software design, "the 'distance' between the design concept and the final product is shorter than in, say, architecture" (p. 93). This does not imply, however, that in those domains design and implementation are not separated. It might, however, clarify results such as our observation that software designers find it particularly difficult to separate design from coding during a design project (Author, 1987).

3.8 Possible forms of evaluation

Domains differ in the means that may be used in order to evaluate design proposals (Malhotra, Thomas, Carroll, & Miller, 1980, pp. 129-130). In engineering, more or less "objective" measures of future artifacts' performance can be used. One can calculate whether a particular design (e.g., a bridge) meets functional requirements, for example, accommodation, and maximum load. Using such measures, different proposals can be ranked somewhat objectively. The results of qualitative evaluation based on subjective criteria such as aesthetics, are much more difficult to translate into a "score", and thus to compare. These are of course extremes; in between are evaluations based on different types of simulation, physical and mental.

3.9 Delay of implementation

Something that has been considered to make the solving of social-science problems particularly difficult, is the "delay from the time a solution is proposed and accepted to when it is fully implemented" (Voss, Greene, Post, & Penner, 1983). "Naturally, a good solution anticipates changes in conditions, but anticipation can be quite difficult". (id.) This remark that is particularly applicable in the social-science domain may also hold for other design areas. The underlying factor may be of influence on solution-proposal evaluation (through simulation and other means).



3.10 Impact of an artifact on people's activity and the possibility to anticipate it

Predicting people's future use of an artifact (procedure, system, building) and further anticipating the impact of the artifact on human activity, is one of the "characteristic and difficult properties" of designing (Carroll, 2000, p. 39). Indeed, "design has broad impacts on people. Design problems lead to transformations in the world that alter possibilities for human activity and experience, often in ways that transcend the boundaries of the original design reasoning." (Carroll, 2000, p. 21)

Even if all design has impact on people, certain domains seem more sensitive than others are. HCI, that is the domain with which Carroll (2000, p. 39) is especially concerned in his discussion quoted above, is an example of a domain in which design has particularly broad impacts on people. Yet, this holds for all design with social implications.

A related issue is the possibility to anticipate the impact. This possibility may vary between domains, but not necessarily in the same form as the degree of impact. It depends, amongst others, on the possibility to simulate the artifact, or to test it in other ways.

3.11 The transformative nature of the artifact

This dimension is a specification of the previous one. The impact of certain systems on people is of a "transformative" nature (Carroll, Rosson, Chin, & Koenemann, 1997, p. 63): these systems "fundamentally [alter] possibilities for human behavior and experience". An example is systems that enable novel educational activities, such as the artifacts developed by Carroll and al.

3.12 The role of the user in the design process

In every design domain, users are central—even if not always for the designers. Artifacts are to be used by people, even if this use can be more or less direct. Domains differ, however, with respect to the way in which they take into account the potential, future users and their use of the artifact. In design of HCI, for example, there has been much effort towards the integration of user data into the design. This has varied from such data being introduced into the design by design participants who "know" the users, but are not these users themselves, to approaches such as participatory design in which the users have themselves a voice in the design process (Carroll, 2002).

It seems likely that the number and variety of participants who take part in a design process influence this process, probably more its socio-organizational than its cognitive aspects. Yet, on a cognitive level, the difficulty of integration may augment with the number of perspectives, and thus of representations, to be integrated. In addition, the participation of "non-technical" design participants may introduce a specific difficulty—again not only of cognitive nature—both for the users and for their professional and "technical" design "colleagues".



3.13 Artifacts' behavior over time

According to Löwgren (1995, p. 94), "interactive systems are designed to have a certain behavior over time, whereas houses typically are not". Even if this assertion is questionable with respect to the "behavior over time" in general of interactive systems and houses, artifacts indeed differ with respect to this behavior—and the types of behavior over time that different artifacts may display or undergo are also quite diverse. An artifact's behavior over time may be related to its possibly "transformative" nature, to its use by people who are not necessarily transformed by this use, to its deterioration, or to people's interaction with the artifact, as referred to by Löwgren, to name just a few sources of variability for artifacts' behavior over time.

Houses may not display "behavior" over time, but they change. Systems such as organizations or interactive software systems are subject to quite other types of change. "Good" designers anticipate the transformation that the artifacts that they design may undergo—be it of an (inter)action, evolution, or deterioration type. For social and/or interactive artifacts, this anticipation may be performed through simulation. The behavior of certain technical systems or other engineering artifacts may be anticipated based on calculations.

3.14 software design and design of HCI

This title translates a search for a dimension: software design and design of HCI are often considered as more or less, or even completely different from design in other application domains. In fact, the resemblance between the two is also an object of discussion. In the cognitive design research literature, one frequently encounters allusions to, or implicit testimonies of the specific character of software design compared to other types of design—design of HCI is much less the object of discussion in this context. The responsible dimension—or dimensions—remain(s), however, unexplored (see Author, 2006, for a more detailed discussion of the potential specificities)

3.15 Other dimensions

There are, however, also authors who consider that software design is comparable to other design tasks. In his paper *How is a piece of software like a building? Toward general design theory and methods*, Gross (2003) advances that pieces of software and buildings are alike on several dimensions. Not one of the dimensions mentioned by Gross appears among those proposed above. They may constitute still other dimensions on which forms of design may differ. Gross (2003) quotes the *size* of software and buildings; their *level of complexity*; the *type of use or user*, which may change (more or less), or may remain constant; the degree to which components of the artifact may be *subject to change or renewal* (cf. an artifact's behavior over time); their *lifetime* (that may be more or less extended); the proportion of *reusable components* in the artifact's structure; the *difference or equivalence between client and user*, and the *sanitary risks and safety concerns* that particular uses or states of the artifact may introduce. According to Gross, pieces of software are like buildings on all these dimensions.



4. Conclusion

The dimensions presented in this paper constitute a list, presented without any internal organization. It is, however, conceivable that not all dimensions have the same degree of influence of the design activity. Given our view of design as the construction of representations, we might suppose that dimensions related to representational structures and activities (see §3.2 and 3.3) are more influential than other ones. However, the dimensions listed may depend on other underlying factors and their influence on the activity may exert itself by way of representational structures and activities.

We started this paper, noting explicitly that we adopted a descriptive, not a normative perspective. The dimensions—if indeed they are factors influencing the design activity—and the characteristics of the ensuing activities may have their methodological and other implications for design support. Given the centrality of representation in designing, the development of appropriate support modalities for representational activities and structures imposes itself. However, according to the role of representation, and the type of representation preferentially used in design tasks in specific application domains, the development of specific support modalities may be worthwhile. Research on these questions may, interestingly enough, that is in spite of the probable specificity of software and HCI design, take advantage of the progress already obtained in these two application domains. Indeed, there has been considerable research on visualization, and visual programming languages in the domain of software and HCI design, for example on diagrammatic reasoning (see the diagrammatic reasoning site, retrieved 19 October 2005, from <http://www.hcrc.ed.ac.uk/gal/Diagrams/>; see also Blackwell, 1997). More generally, not restricted to the domain of design, there is potential useful research into representational formats and their exploitation (e.g. in research on multiple—external—representations, see van Someren, Reimann, Boshuizen, & de Jong, 1988).

Our list of candidate dimensions that might differentiate "forms" of design is a start for their further elaboration and analysis. A next step would be to elucidate if indeed, and if so how these—and other—differences influence design activity and its result, the artifact. Still another step—announced in the introduction of this paper—concerns examination of the fourth point of our hypothesis, namely that in spite of the possibly different forms of design, design remains a specific cognitive activity relative to other cognitive activities.

References

- Alexander, C. (1964). *Notes on the synthesis of form*. Cambridge, MA: Harvard University Press.
- Archer, L. B. (1984). Systematic method for designers. In N. Cross (Ed.), *Developments in design methodology* (pp. 57-82). Chichester, United Kingdom: Wiley. (Originally published by The Design Council, London, 1965)



- Blackwell, A. F. (1997). *Diagrams about thoughts about thoughts about diagrams*. Retrieved 19 October, 2005, from <http://www.cl.cam.ac.uk/~afb21/publications/AAAAI.html>
- Carroll, J. M. (2000). *Making use. Scenario-based design of human computer interactions*. Cambridge, MA: The MIT Press.
- Carroll, J. M. (2002). *Dimensions of participation. Elaborating Herbert Simon's "Science of Design"*. Paper presented at the Workshop "User-Centred Design" of the International Conference In Honour of Herbert Simon "The Sciences of Design. The Scientific Challenge for the 21st Century", Lyon, France: INSA.
- Carroll, J. M., & Rosson, M. B. (1985). Usability specifications as a tool in iterative development. In H. R. Hartson (Ed.), *Advances in human-computer interaction* (Vol. 1, pp. 1-28). Norwood, NJ: Ablex.
- Carroll, J. M., Rosson, M. B., Chin, G., & Koenemann, J. (1997, 18-20 August). *Requirements development: Stages of opportunity for collaborative needs discovery*. Paper presented at the Conference on Designing interactive systems: processes, practices, methods, and techniques (DIS'97), Amsterdam.
- Colleague, Author, & Colleague (2004). Title.
- Chandrasekaran, B. (1983). Towards a taxonomy of problem solving types. *AI Magazine*, 4 (Spring)(1), 9-17.
- Clancey, W. J. (1985). Heuristic classification. *Artificial Intelligence*, 27, 289-350.
- Cross, N. (1986). Understanding design: The lessons of design methodology. *Design Methods and Theories*, 20(2), 409-438.
- Cross, N. (Ed.). (1984). *Developments in design methodology*. New York: Wiley.
- Goel, V. (1994). A comparison of design and nondesign problem spaces. *Artificial Intelligence in Engineering*, 9, 53-72.
- Goel, V., & Pirolli, P. (1989). Motivating the notion of generic design within information-processing theory: The design problem space. *AI Magazine*, 10(1), 18-36.
- Gross, M. D. (2003, November 2-4). *How is a piece of software like a building? Toward general design theory and methods*. Paper presented at the National Science Foundation (NSF) Invitational Workshop on Science of Design: Software Intensive Systems, Airlie Center, Virginia.
- Hubka, V., & Eder, W. E. (1987). A scientific approach to engineering design. *Design Studies*, 8(3), 123-137.
- Jones, J. C. (1984). A method of systematic design. In N. Cross (Ed.), *Developments in design methodology*. Chichester, United Kingdom: Wiley. (Originally published in J. C. Jones and D. Thornley (Eds.) (1963). *Conference on design methods*. Oxford, United Kingdom: Pergamon)
- Lee, A., & Pennington, N. (1994). The effects of paradigms on cognitive activities in design. *International Journal of Human-Computer Studies*, 40, 577-601.
- Löwgren, J. (1995). Applying design methodology to software development. In G. M. Olson & S. Schuon (Eds.), *Symposium on Designing Interactive Systems. Proceedings of the conference on Designing interactive systems: processes, practices, methods, & techniques (DIS'95)* (pp. 87-95). New York: ACM Press.
- Malhotra, A., Thomas, J. C., Carroll, J. M., & Miller, L. A. (1980). Cognitive processes in design. *International Journal of Man-Machine Studies*, 12, 119-140.
- Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall.
- Scaife, M., & Rogers, Y. (1996). External cognition: How do graphical representations work? *International Journal of Human-Computer Studies*, 45(1), 185-213.
- Science of Design (SoD)*. (2004, September 6, 2005, last update date). Retrieved October 12, 2005, from <http://www.nsf.gov/pubs/2004/nsf04552/nsf04552.htm>
- Thomas, J. C., & Carroll, J. M. (1979). The psychological study of design. *Design Studies*, 1(1), 5-11. (Also in Cross, N. (Ed.). (1984). *Developments in design methodology* (pp. 1221-1235). Chichester, United Kingdom: Wiley.)
- Ullman, D. G., Dietterich, T. G., & Stauffer, L. A. (1988). A model of the mechanical design process based on empirical data. *AI EDAM*, 2, 33-52.
- van Someren, M. W., Reimann, P., Boshuizen, H. P. A., & de Jong, T. (1988). *Learning with multiple representations*. Amsterdam: Elsevier.
- Author (1987).
- Author (2003).
- Author (2004).
- Author (2006).
- Author, & Colleague (1990).
- Voss, J. F., Greene, T. R., Post, T. A., & Penner, B. C. (1983). Problem-solving skill in the social sciences. In G. Bower (Ed.), *The psychology of learning and motivation* (Vol. 17, pp. 165-213). New York: Academic Press.
- Zhang, J., & Norman, D. A. (1994). Representations in distributed cognitive tasks. *Cognitive Science*, 18, 87-122.