How designers work - making sense of authentic cognitive activities

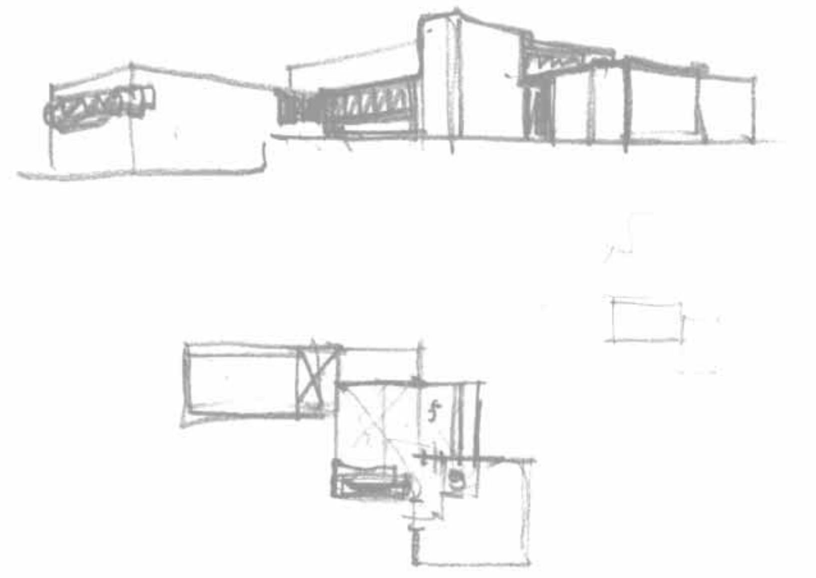
Gedenryd, Henrik

1998

Citation for published version (APA):
Gedenryd, H. (1998). *How designers work - making sense of authentic cognitive activities*. [Doctoral Thesis (monograph), Cognitive Science]. Cognitive Science.

Total number of authors:
1

How designers work



How designers work

Henrik Gedenryd

Printed in Sweden by Jabe Offset AB, Lund, 1988. For further information, corrections, and updates, see the author's web page at *http://lucs.fil.lu.se/people/henrik.gedenryd*

ISBN 91-628-3210-7 — ISRN LUHFDA/HFKO-1007-SE
ISSN 1101-8453 — Lund University Cognitive Studies 75

Table of contents

Acknowledgements

I would like to express my gratitude to the Knut and Alice Wallenberg foundation for the grant that financed my visit to the University of California, San Diego, in 1995–1996. I also wish to thank the Cognitive Science department, and especially Edwin Hutchins who was my host there. I shared an office with Brian Hazlehurst, and our discussions gave me invaluable insights into for example anthropology, which I didn't realize at the time.

The academic environment at Kognitionsforskning in Lund has been an exceptionally creative one, and it is of a very high international standard. I would like to thank everyone who has taken part in making it so, and in our seminars, with countless discussions of theoretical terms, explanations, and other topics.

Designkollegiet, the Swedish interest group in design research has also been very valuable for me, and I hope it will live on in the future. I would like to dedicate this book to the memory of Donald Schön. The extent of his influence on my work will become apparent on the following pages.

Credits for cited pictures and opening quotations

For some cited figures and opening quotes I have collected the reference information here instead of inside the book: Cover image, from Dreyfuss 1955; page ii from Tegethoff, Dyckes & Mies van der Rohe 1985, reduced to 80% of original size to fit page. Opening quotation in chapter 1, attributed to Beatrice Ward by Spiekermann & Ginger 1993. Flowcharts in figure 1.1 are from Jones 1970, pp. 76, 78; figure 1.16 is from *ibid.*, p. 50. Figures 1.4 and 3.2 from Dix *et al.* 1993, pp. 149, 158. Figure 3.1 and sketches for Quist and Petra episode in chapters 3 and 4 are from Schön 1983, pp. 82–87. Tetris figures 5.1 and 5.2 adapted from Kirsh & Maglio 1992. Sketches in figures 6.1–6.3 cited from Arntson 1993, pp. 9, 6, 7. Drill prototypes in figure 6.4 are from Heskett 1980, p. 168; elevator diagram in 6.5 from Guindon 1990b, p. 286. All copyrights are those of the original owners.

0. Introduction

0.1 Making sense of design as a cognitive activity

Design

During recent years, the scientific interest in design processes has grown rapidly. The working processes of designers have been well documented in a number of domains: alongside the classical design domain, architecture, there are also for example industrial design and graphic design, which today are established and recognized design disciplines. In addition, such fields as interaction design and information design are emerging and trying to establish their own identities.

From the studies of these various domains, the knowledge of authentic design processes is increasing. There is also an elementary understanding of *why* they work. However, this is still mostly a list of miscellaneous studies and their assorted insights; the work remains to find how each of the existing pieces can be put in relation to each other, as parts of a bigger picture.

At the same time, there already exists a "received" theoretical perspective on design. Known as *design methodology*, it is based on logic, rationality, abstraction, and rigorous principles. It portrays, or rather *prescribes*, design as an orderly, stringent procedure which systematically collects information, establishes objectives, and computes the design solution, following the principles of logical deduction and mathematical optimization techniques (cf. Alexander 1964, Asimow 1962, Jones 1970, Simon 1981). This view is still very much alive (as is evident in e.g. Dasgupta 1989), and there is good reason to believe that this won't change for a long time.

However, discontent with this approach is widespread and quite old, even though no substantive replacement has yet been proposed. Experience from design practice and from studies of authentic design processes has consistently been that not only don't designers work as design methodology says they should, it is also a well established fact that to do design in the prescribed manner *just doesn't work* (Alexander 1971, Broadbent 1973, Lawson 1980, etc.).

Cognition

There is a similar situation in cognitive science. The conventional theories are highly sophisticated, most stringent in their form, and written in the language of computer science, mathematics, and formal logic. They were developed to explain the most advanced of intellec ctive reasoning, and so forth.

And also in this area, there is a growing body of work on making sense of authentic cognitive activities beyond psychological experiments and computer simulations. These activities are often of an everyday nature and so may appear mundane. Nevertheless, for those who look closely enough, they hold great sophistication underneath the surface, and they perform their functions very well. However, as also these authentic activities call for explanations that are substantially different from what the prevailing theories can provide, they too remain to be properly accounted for.

At the same time, the conventional, proposed cognitive mechanisms are becoming increasingly questioned, as their limited achievements have not been able to match the great claims that have been made on their behalf. Therefore the apparently mundane activities are increasingly being revaluated, as they in spite of their plain looks manage to perform the functions which the sophisticated models have failed on.

The gap between the ideal and the actual

There are in both fields a received theoretical perspective, based on idealized views of rational behavior; chapter 1 serves to show how very closely related the two perspectives are. But the received theories are increasingly being called into question in both fields (cf. chapter 2). At the same time, there is an accumulating mass of evidence on design processes and cognitive performance under authentic circumstances. However, there is still an absence of a real theoretical alternative that can account for this growing body of knowledge.

The problem is in both fields a discrepancy between the received, theoretical views of how things ought to work, and how they have turned out to work in reality—a gap between the *ideal* and the *actual* which needs to be filled with a new explanation, a theory of human performance in these authentic activities. The aim of this book is to present such an explanation (chapters 3 to 6).

The promise of this topic lies in the many commonalities between the two fields, which hold the promise of a synergy. Not only in the gap that they share, but also in the building blocks that each field has already produced. By combining these building blocks, one can gain leverage in building a single, joint explanation for both domains.

From should to making sense

It seems that the shared gap has come out of a similar line of development in both disciplines: The "ideal" approach found certain kinds of solution formally elegant and powerful, therefore assuming that this was how things "had to be". And these *a priori* assumptions were made so confidently that there was hardly a need to ask if they worked, much less to test whether they did—and if they didn't at first, then they surely would if only given some refinement.

But some have found it increasingly hard to ignore the disparity between promises and what has so far been delivered. They have also recognized the procedures that people actually use, and the merits of these procedures in particular. There has therefore been a shift from laying down how things ought to be, toward an increasing estimation of authentic practice, and an interest in *making sense* of the sophistication that is inherent in the mundane.

And this is where the gap resides, in the vacuum between the existing, ideal theories, and the desire to make sense of authentic design activities and cognitive processes. In each field there are the beginnings of this; so far, the design side has come a little further along the way, but also there, the work of filling the gap between *should* and *making sense* is still very much in progress. To fill this gap that exists in both fields is the purpose of this book.

Hence, the need for theory is still great, to explain and make sense of design as a cognitive process, but also to "give designers reason" for what they do, so that practice is no longer looked down on as imperfect and irrational, but is acknowledged for its merits. If so, practice may perhaps take advantage of an improved understanding of its underlying principles.

Case in point: The renaissance of sketching

A good illustration of these developments is the devaluation and subsequent revaluation of sketching. When design research began, it was practically beyond discussion that what was then known as "design-by-drawing" was inadequate as a means for modern-day designers. This inadequacy was even considered the primary reason for developing new procedures of designing, which led to the birth of what would become design research:

> The writings of design theorists imply that the traditional method of design-by-drawing is too simple for the growing complexity of the man-made world. This belief is widely held and may not require any further justification. (Jones 1970, p. 27)

At this point, no distinction was made between working sketches and the final, carefully performed production drawings. Eventually there would be the occasional reference to the nature of sketching, such as those by Rittel (1972) and Graves (1977), although it was probably Schön (1983, 1987, 1988, 1992, Schön & Wiggins 1992) who started the revaluation in earnest, leading to the wide interest in sketching today (e.g. Goel 1995, Herbert 1993, Lawson 1980/1997).

The developments that have since taken place are reflected in the editions of Lawson's *How Designers Think*: The original edition is from 1980; the second from 1992 has a new chapter on designing with computers; in 1997, the third edition added a chapter on sketching with paper and pencil!

This is of course quite ironic, but it should also be seen in relation to the backlash that has occurred regarding computer-aided design, which like the new methods is still advocated as being indisputably superior to earlier procedures. When the limitations of computer tools became apparent, this contributed to the reappreciation of sketching. But the 180 degree reversal on sketching since the axiomatic rejection of "design-by-drawing" has also come from the turn away from the new design methods. The sophistication of paper and pencil became noticed only when the computer aids and new methods proved to lack the capacity that these decidedly low-tech tools had (cf. Black 1990). These developments are a prototypical example of science driven by ideals, and its shift into the sense-making approach. And central as it is, several aspects of sketching will be treated in detail in this book, from chapter 3 and forward.

A general trend toward making sense

I perceive a general scientific trend in this kind of revaluation and in what I will call *making sense* of authentic human activities; a trend which has been slowly emerging for some time now, in diverse areas and fairly independently. Its shared aim is to understand these activities better: the performance by which people accomplish them, and the human abilities that enable them to do so. Especially, to understand *why* they are performed as they are, which is particularly important since the infallible theories have repeatedly proven incapable of what has been claimed of them. In contrast, people's outwardly simple means have repeatedly shown to work where the "ideal" methods have failed. They have often proven to be superior, if only understood on their own proper terms, not by some other inappropriate standard. Exactly this has been the case with sketching.

In design, another very good example is Guindon's account of so-called "opportunistic" design practices: from having previously been regarded as failures and deviations from correct behavior (e.g. Adelson & Soloway 1985), to being superior to the prescribed structured methods (1990a, etc.). Furthermore, there is a large number of highly useful design techniques that have been revalued in the same way, and of which sketching is only one. So-called "low-fidelity" prototypes (Rettig 1994) are one case in point; I will address a number of these techniques in chapter 6.

In other domains, probably the most important case is the reassessment of spoken language and conversation, which have long been viewed as a corrupted form of proper, written language; reified in the Chomskian competence vs. performance distinction. The epitome is perhaps the demonstration by Schegloff, Jefferson & Sacks (1977) of how "errors" in spoken language are better understood as "repair" than as signs of cognitive limitations, and thereby as an effective aspect of normal, equally effective conversation patterns, rather than as a poor derivative of written language.

Outside language, for example Hutchins (1980) has showed that the reasoning in Trobriand land negotiation is quite sensible even though it does not follow Western principles; later he did the same regarding Micronesian navigation techniques, which had previously been deemed as useless and based on native superstition and folklore (1983, 1995). A quotation reflecting the earlier view, and Hutchins' comment on it, reflect the two attitudes:

> Polynesians and Micronesians accomplished their voyages, not thanks to, but in spite of their navigational methods. We must admire them for their daring, their enterprise and their first rate seamanship. (Åkerblom 1968, p. 156)

Hutchins comments:

> I hope this chapter succeeds in laying such notions as Åkerblom's to rest. In fact, it seems more likely to me that we who have stud-

> ied Pacific navigation have accomplished what understanding we have, not thanks to, but in spite of our own cultural belief systems. (Hutchins 1983, p. 224)

Elsewhere, Lave (1988) demonstrated the logic and rationality in everyday problem solving, in spite of its violating the principles of logic and rationality. Norman (e.g. 1988) made sense of people's actions in several incidents in aviation and automation, e.g. the Three Mile Island incident, attributing blame to flawed instrumentation, rather than to "human error" as is the standard procedure.

These are but a few examples of *making sense,* giving people good reason for doing what they do, when they don't do what science and engineering have dictated. After a century (at least) of measuring people by the standards of formal logic, mathematics, engineering, statistics, and so forth, we are slowly beginning to measure them on their own terms. Doing so includes the task of understanding what these terms *are*. If this approach rests on any assumption, then it is to grant people that they *do* make sense; the researcher's task is then to construe *what this sense is* that they make, not taking a certain approach for granted, whether it is because it has some desirable formal properties, or because no one considered any other option.

Perhaps the interpretation of foreign cultures was first to come up against this problem of finding the adequate standard of measure, or *yardstick* as I shall call it; only later would it be realized that the human being is not a member of the culture of formalistic, rigorous and stringent logical principles, and should not be measured within this cultural framework, but that we must first identify its own principles. It is probably no accident that the sense-making approach was pioneered by people from anthropology and related fields who entered into linguistics and cognitive science, as in the above examples.

0.2 Two views of cognition: Intellectual or practical?

As well as there being as there is a trend toward making sense of human activity, there is also a trend in how the sense-making turns out to portray human performance. In common-sense terms, this can be described as a transition from a view of cognition as basically *intellectual*, to one where it is instead conceived of as *practical* by nature. This trend can also be observed in models of design activity. The theoretical argument of this book can be said to revolve around the transition from an intellectual to a practical mode of cognitive explanation.

The traditional accounts have been based on an idealized norm for what cognitive performance should look like, heavily influenced by the intellectual ideal of high intelligence and abstract reasoning. In contrast, the sense-making accounts seem to unanimously paint a picture where such intelligence and abstract thinking play a quite different role. Instead, cognitive performance seems to rely on faculties that look quite mundane and primitive when seen through intellectual eyeglasses. Here it should be kept in mind, however, that also those feats thought to require and display abstract reasoning have been found to exhibit practical and purportedly primitive patterns.

From ideal to intellectual to intramental

The "ideal" mind-set, and the view of cognition as basically intellectual, made up the very essence from which cognitive science was born. The precursors were located in formal logic and proof theory (e.g. the work of Turing), bounded rationality (Simon 1947), and so on. This early work was also intimately connected with the theoretical foundations of computer science (e.g. Chomsky 1957). When the mind was studied empirically, the chosen tasks were considered indicative of high intelligence, such as chess, logical deduction, mathematical problem solving, or tasks resembling IQ tests. An account straight from the horse's mouth can be found in the historical addendum in Newell & Simon (1972).

This strong bias naturally meant that theories also came to focus on the same kind of cerebral tasks. Thus, in this view, *intellectual* tasks and abilities—abstract thinking—were regarded as the prototypical kind of cognitive activity, i.e. which theory took as its first priority to explain. Accordingly, more mundane everyday activities, and the components of action and interaction with the world that they involve, were considered of secondary importance to cognition (and to cognitive theory). The rationale was that once the more difficult problems could be handled, then the simpler ones would easily follow—a sensible conjecture at the time.

From this focus, the influence of cerebral abilities on the theories that were developed became very strong, whereas cognitive abilities involving non-mental functions were strongly underrepresented. I will refer to these as theories of *intra-mental* cognition. With this I indicate the view of cognition as a process that is contained entirely within the mind, and which is performed by the mind alone, cognition being strictly isolated and separated from action, perception, and every aspect of the surrounding world, be it material, social or cultural. In this view, the study of cognition is often defined as the study of mental processes, and the two are thus considered equivalent.

In summary, there is a natural link from the ideal view of cognition, via the emphasis on intellectual activities, to the view of cognition as intramental. This position will be treated in detail in chapter 1.

Problems with the intramental view

The intramental focus would not have been a problem if only the resulting theories could also be made to account for other types of activity. However, whereas they have been able to account for intramental abilities and processes in a simple and natural manner, explanations of non-cerebral activities have been significantly more belabored and less convincing.

The exact reasons for the failure of intramental theories to account for factors outside the mind remain to be fully understood, but it seems (although this is still widely contested) that the focus on the isolated mind has resulted in a skewed conception of the tasks that were studied (related to the issue of ecological validity), which in turn came to give the theories deeply seated peculiarities that prevented such a satisfactory extension (cf. Hutchins 1995).

The most striking example of a not-so-convincing explanation is how the cognitive role of the physical world has been accounted for, by treating the world as an extension of long-term memory, while at the same time considering long-term memory as a part of the environment (Newell & Simon 1972, Simon 1981). But however bizarre this explanation appears, there seems to be no better alternative, given the foundational assumptions of these theories. Even more interesting is that the proponents of this view see no problem with this explanatory approach. In fact, this remains the official explanation to this day (see e.g. Goel 1995, Larkin 1989, Larkin & Simon 1986, Newell, Rosenbloom & Laird 1989, Vera & Simon 1993).

Historically, cognitive science began with theories of pure thought and intellectual activity, to which interaction with the environment was added later, e.g. in Newell & Simon's (1972) theory of problem solving, and the model of planning in the work of Miller, Galanter & Pribram (1960), but for which no proof of concept has ever been provided. I believe the question of what caused this failure to be an important one, which must be settled before we can put these theories and their hidden assumptions behind us, so as to make progress and not make the same mistakes again. This is also necessary if we are ever to move beyond general criticism and vague claims that "one must also take social and cultural factors into account", etc. For example, it is quite widely recognized that Information Processing Theory (Newell & Simon 1972) has proven inadequate as a fundamental theory of cognition. However, the standard diagnosis is that its problems are located in the computer model of mind and the symbolic view of cognition. As will become evident, much of this book revolves around the question of exactly *what* caused the problems and rendered these theories deficient. As I will argue in detail, although these features are problematic, they are not the crucial defects.

The heart of my diagnosis, which I will present in chapters 1–3, is that *intramentality* is the culprit. Even among those who reject information processing theory, the view of cognition as *pure thinking*, which I consider to be the central problem, remains widely unchallenged in the belief that symbols and so on are the real problem.

I believe this is much because the intramental character hasn't been explicitly stated or advocated to the same extent. In comparison, Newell & Simon expressly described their theory as symbolic and computational, and these aspects have also been discussed by many others. The bottom line is that if only the usual suspects are charged and discharged, then the real culprit remains at large, continuing to cause the same trouble as before.

Accordingly, when I refer to "traditional" cognitive theories, or "conventional" cognitive science, etc., it is by the criterion of *intramentality* that I define these terms. This is a small but crucial shift from the current view where the defining characteristics of "traditional" cognitive science are held to be precisely the computer model of mind, symbols, and information processing theory. My shift in reference is of course based on the point that intramentality is the problem, and not these factors.

From practical to interactive

In contrast to the intellectual perspective, there is an alternative approach where cognition is seen as fundamentally *practical* by nature. Accordingly, in this view practical activity is considered more fundamental to cognition, and so, the theory gives a higher priority to explaining practical activities and abilities.

Some philosophical precedents

The contrast between the intellectual and practical positions is reflected in a number of prominent historical debates of 20th-century science and philosophy, the issues of which will at least implicitly reappear in what follows. Early among these, beside the pragmatists, are the respective views of Husserl (1900/1970), who based his work on the phenomenology of mathematics, and Heidegger (1927/1962), who countered him by arguing that non-reflective being is the fundamental mode of existence, and abstract thinking and reflection being the result of a disturbance, causing you to be "thrown" out of the basic mode of just doing (also cf. Dreyfus 1991).

Later there would be the contrasting views of knowledge as linguistic/propositional and explicit vs. knowledge as tacit/implicit and based in practical activity (Dewey 1925, 1933, Polanyi 1958, Schön 1983). These are also referred to as *knowing that* vs. *knowing how*; which are clearly the intellectual and practical views of knowledge, respectively. The antagonists personified in the early and late Wittgenstein should also be mentioned. Again, this list is by no means exhaustive. As part of the turn toward the practical dimension there is of course a revaluation of non-intellectual activities as being "worthy" of scientific attention; cf. the discussion of "making sense" above.

The evolutionary perspective

Beside the ability to explain cognition better, the choice of giving theoretical priority to non-intellectual cognitive abilities is often motivated with evolutionary arguments: the skills that are uniquely human have developed from the more fundamental capacities that we share with other species. A second tenet is that these "lower" capacities are more powerful, and play a more important role, than intuition tells us, even in those tasks that we instinctively tend to consider simply as "thinking".

This goes counter to the intellectualist take on evolution, which holds that humans, unlike lower species, have a monolithic mental module defined in their genes, which gives us all the intramental abilities that make humans special: language, propositional attitudes, problem solving, and so on (cf. Anderson 1983, Donald 1991). The problem with this view is of course that it places a magnificent burden on evolution, in requiring this mammoth structure to have appeared out of nowhere in humans, with no intermediate evolutionary forms, which is otherwise virtually unheard of in evolutionary biology.

The practical view can cut the "uniquely human" stuff into smaller chunks, seven or so, and distribute parts of the explanatory burden from genes and intramental faculties onto material and immaterial human culture, yielding these abilities *in co-evolution with* species-inherent genetic progress. The result is an account that is much easier to swallow, not only in evolutionary terms, but also in terms of how cognition could attain the human capacities.

The practical view of cognition

In a scientific context, the difference between these positions mainly concerns what theoretical explanations should look like. In the intellectual view, theory should be based on the explanation of thinking and abstract skills of the kind mentioned above, whereas in the practical view, cognitive theory is built around practical skills and authentic activities. These serve as the basis for explaining intellectual abilities as well, which are also seen as essentially practical, albeit refined, skills which remain dependent on action and the physical world for their operation (think pen and paper—also cf. e.g. Agre & Chapman 1987, Hutchins 1995). Also, they are at least partly of cultural origin, and to some extent acquired rather than innate. Hence, here it is instead *intellectual* abilities that are considered as "specialty cognition" and that are explained in terms of practical skills, instead of the other way around. This is what it means for either kind to be considered fundamental.

Redefining cognition

A major share of this book is devoted to presenting a cognitive theory based on this point of view; I can here merely give a hint of what it will look like, and I will do this in terms of a contrast with conventional, intramental theory. Speaking in general terms, the theory amounts to not defining cognition as narrowly as just thinking. As a result of the historical emphasis on a narrow view, we have come to a point where today many find it hard to imagine how cognition could include something other than strictly mental activities, and what this would then signify; this is an issue that I will also discuss in later chapters.

The extended ontology of cognition

The extended view can be described along two dimensions: one extends cognition to involve other *entities* than the mind, the other spans a wider set of *activities* than merely intramental processes. The first, material dimension is the more tangible one. Here I will claim that not only the mind but also action and the physical world have roles in cognition. Schematically, interaction realizes the link between mind and environment in the cognitive process. The two views of cognition can be represented diagrammatically:

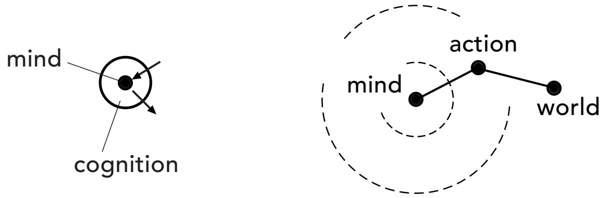


Figure 0.1 Simplified schemas of the traditional, sharply delimited view of cognition as an intramental process, and the wider and less distinctly circumscribed view of cognition adopted here.

The activity dimension

However, this extended "ontology" of cognition is not as significant as the less tangible dimension of *activity*. The most consequential statement of this book is that cognition shouldn't be regarded as thinking (a mental process), but as an activity of *inquiry* (derived from, but not a faithful replica of, Dewey's theory of inquiry, e.g. 1929, 1938, 1949, also cf. Schön 1983, 1987). The most important idea is that there are various activities that are not intramental but which nevertheless have a partially or predominantly cognitive function—or more correctly, that *most* activities have such a cognitive function, as well as having the physical effects which we normally associate with action.

In such activities, action and physical materials are necessary parts, which make these activities possible rather than being the crucial elements in themselves. To make a crude analogy, what makes a pocket calculator more powerful for arithmetic than a pencil is not that it is based on silicon rather than graphite, or that you press plastic buttons instead of moving a piece of wood; these are merely the means which realize the *function* of the calculator. It is this function which is more advanced (for arithmetic). Conversely, it is not graphite that makes pencils superior to computer-aided design for the conceptual stages of design work.

Viewing cognition in terms of function, not ontology

Above all, I advocate that cognition should be defined in terms of function rather than ontology or physical location. In precedence to the extended view in the figure above, cognition should be regarded as the adaptive abilities with which it equips us, rather than as the things that go on in our heads.

This also implies that the entities that are part of cognitive explanations will *vary* with the specific functions that we are explaining. For example, I will be concerned with functions in which the explanations need to include single individuals, more or less, and their activities and working materials, whereas others may study functions that involve, say, multiple actors and artifacts working in concert, as Hutchins has done (1990, 1995). Conceiving of cognition in terms of function rather than ontology means a theoretical shift from realism to instrumentalism that in my view has been long overdue (but which should not be confused with the kind of functionalism that has been popular in cognitive psychology, cf. Clancey 1997). However, this is much harder to represent in a diagram than the three entities of mind, world, and action.

The main fallacy is committed when the unit of analysis chosen entails that cognitive functions are wrongly attributed to the mind instead of other entities (cf. Hutchins 1995, chapter 9). This will give us a theory which mistakenly equips us with superfluous modules because their genuine location isn't eligible for cognitive explanations. This is done for example when something is placed in long-term memory because the environment cannot be made part of the cognitive system. This will yield a view of cognition and the mind which is fundamentally mistaken. The aim is thus not to eliminate the mind from cognitive explanations, but also among other things to yield a better model also of the mind and its function. Such a model is however beyond the scope of this book.

The contrast between the two perspectives and their respective theories makes up the axis around which this book revolves. My argument will concern, on the one hand, the problematic aspects of the conventional view, and an effort to make clear the roots of its problems; and on the other hand, my alternative account, which addresses these problems.

Here, design provides cognitive science with a rich, authentic domain on which to build such a cognitive theory; correspondingly, the cognitive perspective can with such a theory provide design research with an explanation of authentic design processes. This is the synergy which I have tried to engender on the following pages.

0.3 Outline of chapters

The general organizing principle of the book is to move from the existing, ideal-oriented theories successively toward my alternative model and then into its finer points. I first present the conventional theories and dissect them to expose the anatomy of their defects. This is done in chapter 1; the failures are introduced in chapter 2, and the discussion of the conventional theories is concluded at the end of chapter 3. However, before that I begin building my alternative explanation in chapter 3, with the rest of the book going deeper into these issues in chapters 4 to 6, with a brief last chapter that looks back in conclusion at the overall argument, and addresses the wider implications of this.

1. The masterplan

The main topic of the first chapter is a "dissection" and critique of the existing, conventional theories, mainly of cognitive science but also of design research, the result of which lays the foundation for my alternative that is to follow. In chapter ch1 I present the conventional theories in order to disclose their tacit key principles, but also to show that they are in fact *the same principles* in the theories of both cognition and design. As I demonstrate there, these theories and a number of others of the "ideal" lineage are all based on the same underlying model of ideal rationality and ideally rational action. Chapter 1 serves to make this model explicit, and to expose its constituting principles and inherent problems.

2. The general failure of design methods

This analysis then provides the basis for showing *that* these theories fail, in chapter 2, and then *why* they fail, in the last part of chapter 3. The upshot is that the failure of the "ideal" theories is inherent in the underlying model from chapter 1; and that the problematic principles cause the same failures to occur whenever these theories are put to work for their intended purposes.

In the rather brief chapter 2, the purpose is firstly to establish the general and complete failure of design methodology. Since this has been thoroughly documented by design theorists, I only make this point briefly, with reference to their original works. The second purpose is to show that this failure can be traced down to the original geometrical proofs and the domain of formal logic in general. This also goes to show that the failure can be traced to the underlying model of rational action. As the exposition and critique of this model in chapter 1 is rather thorough, much of chapter 2 consists in demonstrating that the principles of the underlying model of rationality indeed are the reasons for its problems.

3. Design and cognition as inquiry

After that, chapter 3 goes on to analyze actual design work, and introduces the fundamentals of my alternative theory in order to capture and explain this. It begins with the contrast between on the one hand the "ideal" view of the problem (or requirements specification) being given already before design begins, and on the other hand that in actual design projects, the task of problem definition amounts to the largest, most difficult, and most important part of the design task.

My alternative draws on the pragmatist view of knowledge, and in particular the theory of inquiry, originating in the work of Dewey (e.g. 1929, 1938, 1949) and then updated and made known in relation to design by Schön (e.g. 1983, 1987). The core idea is that inquiry is an aggregate process with several component functions, *one* of which is action. Hence, in this view cognition consists in inquiry, including all these component functions. It thereby amounts to a composite, physical and concrete activity; this in contrast to the view of cognition as pure thinking, as in the conventional model presented in chapter 1.

At the end of chapter 3, I put the separation of the conventional view in contrast to the compound nature of inquiry, showing that design (and cognition in general) cannot consist in pure thinking, as conventional theory requires. This is why the rational model of action, and the descendant theories of e.g. cognition and design, all fail: abstract thought alone cannot perform the cognitive task required of a designer. The main introduction in this chapter is the elements of the theory of inquiry; in particular, I introduce the notion of a second, "inquiring" or cognitive purpose of action, which will then make up a key element of the theory, and a central concept in the rest of the book.

4. The cognitive roles of action and world

The remaining chapters are concerned with developing my alternative theory and the concepts introduced in chapter 3. The underlying idea in these chapters is of a cognition which comes to include both action and world, with their having cognitive functions and being parts of the cognitive process. Here, sketching is taken as a prototypical physical design activity that is to be explained in this way. The chapter also addresses the question what the roles of these two are; how they *can* have cognitive roles, even, as this extended view of cognition is somewhat counterintuitive.

The explanation I present for the cognitive role of action and world I have tentatively given the name *interactive cognition*. The strategy for this presentation is similar to that of chapter 1: beginning on the surface with the most basic and general issues and then going successively deeper into the finer details. This begins in chapter 4 with a first sketch of what the theory means through an analogy with written and spoken language, where these correspond respectively to intramental and interactive cognition.

5. Interactive cognition

Chapter 5 then goes on to the particulars of the theory. It is framed as an explanation of *why* design activity follows an interactive structure. This is because it *brings important advantages* over working as intramental theory says; since in a sense, interactive cognition works better than an intramental kind would. As in the rest of the book, the theoretical motivation is made on general cognitive grounds, and is not restricted to the domain of design.

This why-argument is presented as four steps, where each represents a certain type of advantage brought by involving (inter)action in the cognitive process; also these steps go from general to successively more narrow and particular. Each consecutive step is made possible by the previous ones, and brings the advantages to a new level; For instance, the first step concerns the advantages brought by addressing the *actual* world instead of dealing with a mental representation—a surrogate—of the world; and the second step adds to that

the ability to employ *action* for cognitive purposes, where this corresponds to the inquiring or cognitive function of action.

6. Making the world a part of cognition

Chapter 6 turns from interactive cognition in itself to the role of the world in this scheme; in doing so, I introduce the notion of a second, inquiring function also of physical materials (i.e. the world). Here I analyze a number of well documented and widely used design techniques that heretofore lack a proper explanation, for instance of why they are so useful; particularly so since they go counter to existing theories. Although the techniques are quite diverse on the surface (e.g. prototypes, scenarios, simulation, storyboards, participatory design), chapter 6 will demonstrate that they can all be explained as serving to *make the world a part of cognition*. Along with this argument, I also analyze what aspects and properties of the world they re-create, and what their contribution to cognition is, and in particular, the *relation* between their properties and cognitive contribution. This then serves as a closer examination of just what the cognitive role of the world consists in.

7. Intermission

In the concluding chapter, I elaborate on the theoretical implications of the techniques from chapter 6: They even go beyond simply using the world in cognition, since they all go to a certain length to *create* working materials that can be given cognitive roles. This means that designers go to some length to even *avoid* having to work intramentally, as the usual theories claim they should do.

This is quite a strong argument against cognition being fundamentally intramental, and thus in support of my extended view of cognition (and design) as inquiry. I thereby briefly address some potential counterarguments from the intramental camp. Hence, I conclude the book by returning to the bigger issues and the debate on whether cognition is fundamentally intellectual or practical. It will however hardly be the last word in this matter.

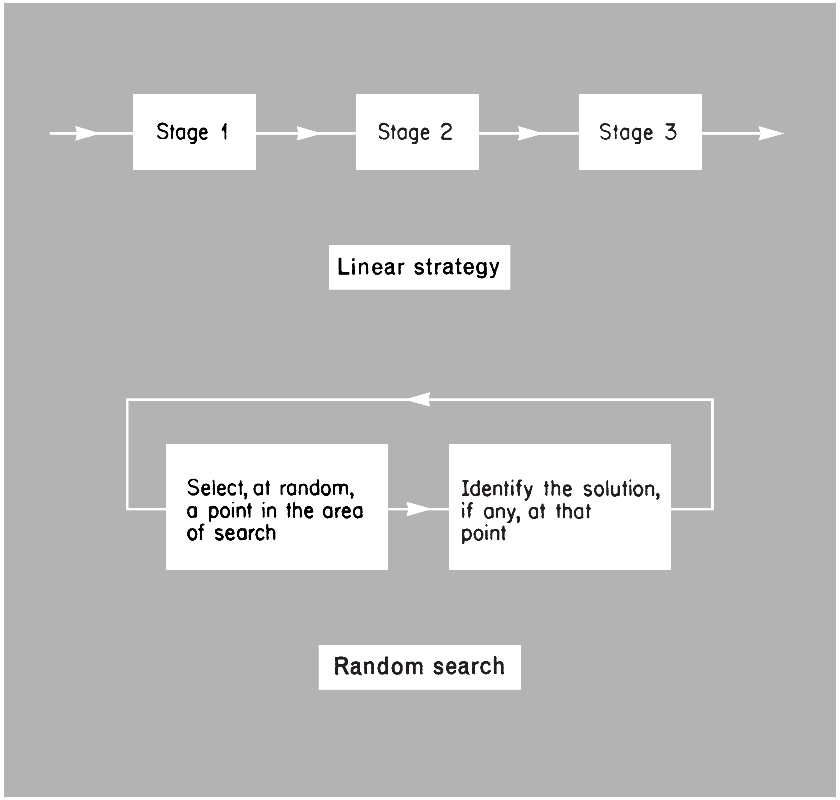


Figure 1.1 Examples of "block diagrams" from Jones 1970.

1. The masterplan

> *No, Watson, this was not done*
> *by accident, but by design.*

1.1 Design methods

When the scientific study of design emerged after World War II, it began as an effort toward developing new procedures for designing. In the face of the increasingly complex tasks that designers were encountering, the pioneers of the field saw a need for improved ways of designing, as they thought the existing procedures were inadequate (Alexander 1964, 1971, Cross 1984, Jones 1970, Rittel 1972). Therefore, the early work almost exclusively sought to develop such new procedures, or *design methods*; and so, the field was appropriately called *design methodology*—the study of such methods. It was also known as "the design methods movement" (Cross 1984).

A design method is a normative scheme that specifies in detail a certain working procedure, the activities to perform, and also a specific order in which the activities should be carried out. It is usually very precise, and the designer is to follow it meticulously. It also covers the design process from beginning to end.

But the easiest way of describing design methods is through the boxes-and-arrows diagrams that always come with them (figure 1.1). The boxes and arrows are always there; it is the labels on the boxes and the connections between them that distinguish one method from another (Jones 1970, p. 61):

> Perhaps the most characteristic feature of the literature on design methods is the prevalence of block diagrams, matrices and networks of many kinds that resemble, to varying degrees, the diagrams and calculations that computer programmers use.

In the history of design methodology, there are two original works that tend to stand out from the rest. They are Alexander's *Notes on the Synthesis of Form* (1964) and *Design Methods* by Jones (1970). Together they epitomize the movement, for a number of reasons. First, they were both rather early and very influential. Earlier versions of the central ideas in both these books were presented at the first conference on design methods in 1962 (Alexander 1963, Jones 1963, Jones

& Thornley 1963). Secondly, even though these two texts and other pioneering work in the field have been fundamentally revaluated since, not least by these two authors themselves, these two works stand out in retrospect as the best exemplars and strongest representatives of the design methods era. Although the ideas in them may have come of age, and other works have long since faded into history, these two have been able to hold their own also as texts, because of the clarity with which they express the central ideas of their original context. Thirdly, because of their scope and depth, and because of the general coherence of the field, these two works together are sufficient to cover the central ideas of design methodology. Jones' book is also very much a compilation of other early work on design methods, ending with a catalogue of various design techniques that had been developed.

An additional work also deserves to be mentioned here. The edited volume *Developments in Design Methodology* (Cross 1984) manages to collect many of the most important papers from the first twenty years of the field, with good introductions to each section. It is therefore an invaluable source. The two previous monographs together with this compilation of classic papers leave little more to wish for as a comprehensive overview of the field.

Four unifying principles

The number of design methods (and accompanying diagrams) that have been published is immense. Probably no two authors have ever agreed on a method, so at least as many methods have been presented as there have been authors. But as people change their minds, the number is probably higher. Therefore, if you begin to review the field and the various methods, you quickly become bewildered by the plethora of variants, the different labels on the various boxes, and the directions of the arrows.

But when you examine a large enough number of variants, patterns begin to form: certain features are due to the specific content of a domain; architecture is different from information design, and so the methods differ. In many cases, different labels disguise the same ideas; and different authors emphasize different aspect of design, so the methods focus on different aspects of the design process. Other variation comes from whether a method is an entirely theoretical construction, or if it has actually been confronted with real design projects, and so forth. (Regarding comparisons between different methods, and their resemblances, see e.g. Jones 1970, p. 24, Lawson 1980, pp. 23–29.) To attempt a comprehensive survey here would thus also be a futile endeavor.

Rather, I will stick to a few prototypical models, and instead concentrate on the common patterns. This is a more viable route, since as much variation as there is on the surface, there is also an Aristotelian essence that the methods share, because they differ in their details rather than in fundamental concepts. To make this essence explicit, I will characterize it in terms of four fundamental principles, which are of particular interest from a cognitive point of view:

1. *separation*: The separation of the design process into distinct phases, with each individual activity being performed in isolation from the others.
2. *logical order*: The specification of an explicit order in which to perform these different activities.
3. *planning*: The pre-specification of an order in which to perform the activities within a phase.
4. *product–process symmetry*: The plan being organized so as to make the structure of the design process reflect the structure of the sub-components of the resulting design product.

These principles do not appear in any design methodologist's lexicon, but they make up the heart of design methods thinking, and give the various methods their family resemblance.

Separation

Out of the four principles, each consecutive one is an elaboration of those before it, drawing out their consequences and filling in their details. From this it follows that they are ordered, from the first being the most general and most fundamental one, to successively becoming more explicit and detailed. Although it may seem abstract and inconspicuous, *separation* is the most important principle, from which the remaining three follow as consequences. The most important separation is to divide the design process into three major phases: analyzing the problem, synthesizing a solution, and evaluating the outcome (Jones 1970, p. 63):

> One of the simplest and most common observations about designing, and one upon which many writers agree, is that it in-

> cludes the three essential stages of analysis, synthesis and evaluation. These can be described in simple words as "breaking the problem into pieces", "putting the pieces together in a new way" and "testing to discover the consequences of putting the new arrangement into practice".

In this chapter, it is the separation of analysis and synthesis that is the most important one. It is the foundation of all design methods, and may well be the most consequential idea of design methodology as a whole. As Jones also indicates, this division was also widely accepted by design methodologists as a basic model of the design process (cf. Cross 1984). Design methods assign such a trivial role to evaluation that it becomes of marginal interest. As Jones here describes evaluation, for example, it seems to be called in only when the real job has already been completed.

Design methods normally make additional separations. In particular, the three major stages are often divided further into several smaller sub-activities.

The principle of separation says that different *functions* of the design process are performed as separate *activities*. With respect to analysis and synthesis, one can say that design activity must serve two functions: understanding the problem and producing a solution. Separation then means that each of these two functions is worked on in a separate phase of problem solving. It is for instance easy to imagine a situation where both of these aspects are worked on together.

Logical order

The second principle concerns the imposition of an *order* among the activities of a design method. Perhaps the distinction between the different activities that a design method is made up of may seem obvious, and the prescribed ordering among the activities may seem more significant. However, even though it might appear so, the working order is a necessity that follows directly from separation, whereas it is not obvious that they should be kept separated: If you do separate analysis from synthesis, then you must perform the analysis before the synthesis, as you have to have to understand the problem before you produce the solution. The same goes for evaluation, it requires that you have something to evaluate, and so must follow synthesis. And conversely, if you do not separate the process into distinct phases then there is nothing to order, so an ordering doesn't make sense. This applies to all other separations that are made: the ordering among the activities is a logical consequence of the purpose that each serves. It is therefore the *logical order*.

Taken together, the first two principles, separation and logical order, generate a basic three-stage model of design; cf. figures 1.2 & 1.4.

Figure 1.2 The basic three-stage design method schema.

Planning

Whereas the logical order concerns the relation between different phases, the third principle aims to lay down the organization of the design activities in even greater detail, to include the activity *within* a phase. Because of the size and complexity of design problems, each of the three major phases is quite complex. Without an internal order, each phase would be a large, unstructured activity, left by the methodologist for the eventual designer to decide. *Planning* consists in setting up a strategy, a plan, for how a particular activity should be performed. The prototypical case is when a plan is set up as the final part of the analysis, and the course of action in the synthesis is thereby laid down before this activity begins.

Product–process symmetry

The fourth principle concerns the decomposition scheme used in the plan; the particular strategy that organizes activity inside the synthesis phase. There is not automatically any *logical* ordering within the phases. Therefore, a decomposition strategy needs to be chosen. This strategy could be *ad hoc*, but typically design methods try to do better than that.

There is however one strategy that is particularly obvious. This is the idea of using the division of the *product* into subcomponents for the decomposition of the *activity* as well: As also the design solution is bound to be complex, it too ought to be broken down into manageable parts. Hence, part of the analysis typically consists in finding such a suitable solution decomposition, usually a hierarchical one. And when you have this decomposition, it is not far-fetched to use it to structure the synthesis activity as well. In effect, the synthesis phase gets a hierarchical organization that mirrors the hierarchical structure of the final product. Hence the process and product are structured in the same way; the decompositon principle consists in a *product–process symmetry*. This lies particularly close at hand since the symmetry results in a natural one-to-one mapping between different parts of the synthesis and of the design product.

All four principles taken together yield a resulting schema that is more complex than the basic three-stage version. As the last two principles are elaborations of the first and second, the complex schema can be regarded as an "elaborated" version of the basic one.

Examples of the elaborated version are the classical "waterfall" model (Boehm 1975, cf. figure 1.3) from software engineering (also cf. Adelson & Soloway 1988, Jeffries *et al.* 1981, Parnas & Clements 1986), and Alexander's (1964) method, which centers on a technique for determining a suitable problem decomposition. These are known as "structured design methods": analysis creates the decomposition structure of the artifact, and which the synthesis is to follow as a "structured decomposition". Together, the basic and elaborated versions capture the central features of most design methods.

To make relations between different design methods stand out more clearly, such as between these basic and elaborated versions, I will hereafter use a "timeline" format which does not obscure these relations, as does the clutter in diagrams of the boxes-and-arrows kind (see figure 1.4).

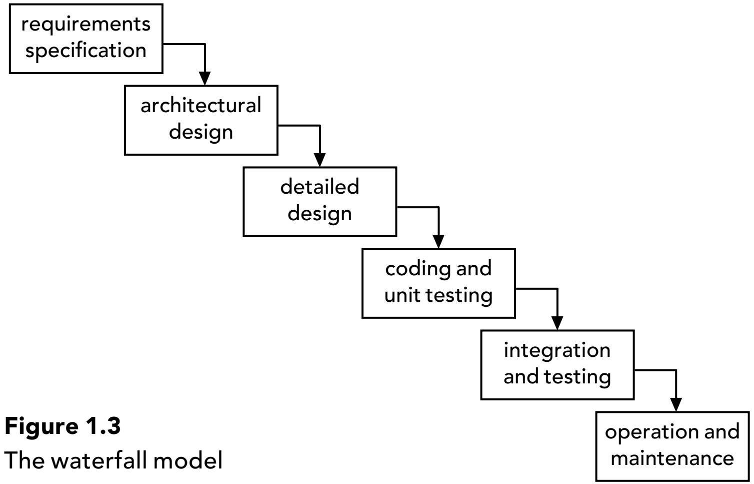


Figure 1.3
The waterfall model
of software engineering.

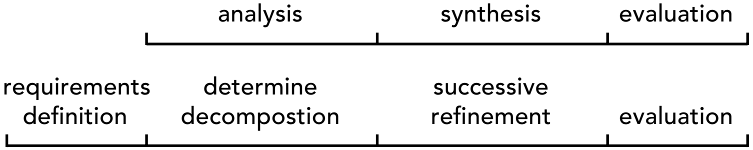


Figure 1.4 Timelines of the basic and elaborated design methods.

However, even with the timelines one problem remains: Different stage models can only be compared approximately, since the internal boundaries between their stages are inherently vague. For example, is requirements definition part of the analysis? Both yes and no are correct answers—mainly because they cannot be held apart in practice. In fact, the greatest weakness of stage models is the principle of separation, as I will argue in chapter 3. It is because such sharp divisions cannot be maintained that the separation of activities breaks down in practice. And for the same reason, one cannot strictly say what goes where within a model, and thus what it corresponds to in another model. For example, analysis in one model sometimes includes the "understand problem" phase of another, while it does not in a third one. Therefore some relations in the timeline diagrams will seem inconsistent.

1.2 The origin of design methods

Why do the design methods look like this? Where do they come from? What is the origin of these methods, the origin of the pattern that is reflected in the four principles? Compared to the motives behind the rise of the design methods movement, the authors of the field have been much less clear about the background of the methods themselves. Rittel (1972) mentions "the ways in which the large-scale NASA and military-type technological problems had been approached" as a major source of inspiration, and also elsewhere there is the occasional reference to general systems theory and operations research. But this does not lead to the answer we are looking for.

Alexander gives some minor, indirect clues when asked about the origins of his (1964) method (Alexander 1971):

> As you know, I studied mathematics for a long time. What I learned, among other things, was that if you want to specify something precisely, the only way to specify it and be sure that you

> aren't kidding yourself is to specify a clearly defined step-by-step process which anyone can carry out, for constructing the thing you are trying to specify. In short, if you really understand what a fine piece of architecture is—really, thoroughly understand it—you will be able to specify a step-by-step process which will always lead to the creation of such a thing. ... So for me, the definition of a process, or a method, was just a way of being precise, a way of being sure I wasn't just waffling.

The step-by-step processes in mathematics that Alexander is referring to are formalized methods for mathematical proofs. It is however Parnas & Clements (1986) who give the most explicit clues—if not any lead to the source in itself:

> Ideally, we would like to derive our programs from a statement of requirements in the same sense that theorems are derived from axioms in a published proof. (p. 251)

Parnas & Clements do not make a big point out of this, or indicate any specific relation to mathematical proofs; these seem to have served more as a source of inspiration to design methodologists, because of their desire to have the same kind of solid foundations for their design choices as mathematicians have in their proofs. Neither did Alexander explicitly model his method after any specific mathematical procedure. His background in the field would rather have provided the "tools" to realize his method.

Hence, the origins of design methods are not well documented. Still, no introductory chapter with any pretensions should be without a reference to the ancient Greeks, and this is where the opportunity arises. Some shallow digging into the origins of logic and mathematical proofs shows a historical influence on design methods that is quite old, but nevertheless still clearly present in design methodology today: contemporary design methods have their roots in the pattern of classical Euclidean geometry proofs. What is more, it is somewhat surprising how many quite different roads in this chapter will be found to all lead back to this same Rome in the end.

Pappus

Or to be more correct, back to Alexandria, as the text in question was written by the Greek mathematician Pappus of Alexandria, probably around AD 300. (The original reference is the Latin translation in Hultsch 1876–77 vol. II, pp. 634–636, first English translation in Heath 1921, see also Polya 1945 and Hintikka & Remes 1974.) In the seventh book of his *Collectio*, Pappus describes what he calls the *analyomenos*, which has been variously translated as "the Treasure of Analysis", "the art of solving problems" and "heuristics". The former is the conventional translation, the latter two are from Polya.

Pappus' text describes a method of analysis and synthesis, to be used for producing geometrical proofs such as those found in Euclid's *Elements.* The crucial elements of mathematical proofs and problem-solving procedures that are presented there stand essentially unaltered to this day. But what is more, the text also contains all the central ideas of design methods; in fact, these amount to the same four principles as those introduced above. The links between design methods, mathematical proofs, and Pappus' original account are remarkably strong, given the vast span of time between them. I will here include an extensive quotation of the original text for two reasons: first, because it has a central place in the argument that follows, and secondly to give enough material to show that the points are valid here even though they have been moved far from the original and very old context (The translation is taken from Hintikka & Remes 1974, pp. 8–9, my italics, and I will depart from the convention and leave out the original text in Greek):

> The so-called Treasury of Analysis is, in short, a special body of doctrines furnished for the use of those who, after going through the usual elements, wish to obtain the power of solving theoretical problems, which are set to them, and for this purpose only is it useful. It is the work of three men, Euclid the author of the *Elements*, Apollonius of Perga, and Aristaeus the Elder, and proceeds by the method of analysis and synthesis.

Analysis traces a path backward from the goal ("what is sought") until you reach the starting point. In geometry this is something given or something already known, e.g. an axiom or an existing proof:

> Now analysis is the way from what is sought—as if it were admitted—through its concomitants in order to something admitted in synthesis. For in analysis we suppose that which is sought to be already done, and we inquire from what it results, and again what is the antecedent of the latter, until we on our backward way light upon something already known and being first in order. *And we call such a method analysis, as being a solution backwards.*

Synthesis goes in the opposite direction from the start through the steps which were found in the analysis, and ends at the goal:

> In synthesis, on the other hand, we suppose that which was reached last in analysis to be already done, and arranging in their natural order as consequents the former antecedents and linking them one with another, we in the end arrive at the construction of the thing sought. And this we call synthesis.

Pappus distinguishes between two kinds of analysis, one for constructing a proof ("theoretical analysis"), and one for ordinary problem solving, i.e. finding and calculating a solution to a stated problem ("problematical analysis"). In the first kind, the proof is the reverse of the analysis:

> In the theoretical kind we suppose the thing sought as being true, and then we pass through its concomitants in order, as though they were true and existent by hypothesis, to something admitted; then, if that which is admitted be true, the thing sought is true, too, *and the proof will be the reverse of analysis.* …

In the second kind, the steps necessary to reach ("synthesize") the solution consist of the steps of the analysis taken backward.

> In the problematical kind we suppose the desired thing to be known, and then we pass through its concomitants in order, as though they were true, up to something admitted. If the thing admitted is possible or can be done, that is, if it is what the mathematicians call given, the desired thing will also be possible. The proof will again be the reverse of analysis.

This concludes Pappus' original text.

Pappus in relation to design methods and the four principles

In relation to design methods, the first thing to note about Pappus' description is of course that he describes a process consisting of two parts having the same functions as in design: analyzing the problem and synthesizing a solution. The division of these two functions into *separate phases* is also there, as is their *relative order*. It is also clear that this order is appropriately regarded as natural or logical, as it is embodied naturally in what the two processes' functions are.

Also the principles of planning and product–process symmetry are present in Pappus' account, albeit more indirectly. This is because geometrical proofs are so very much simpler than design problems; as stated above, these two principles mainly aim to handle the complexity within a phase, a matter which is not as pressing in geometry. For this reason, Pappus provides only a very basic decomposition principle.

The central idea of planning is that of letting the work of the synthesis phase be determined in advance during analysis. As indicated above, this structure may follow different principles. The Pappan approach to this is that the proof consists of the path taken in the analysis, only in the reverse order. The proof is then the plan, as this reverse order is also what the synthesis is to follow.

The product–process symmetry is also present, although this too in a very simple form. The product of a geometrical proof problem is the proof itself: the sequence of steps from what is given to what is sought, which constitutes the demonstration. And in this case both phases, analysis and synthesis, have the structure of the the proof.

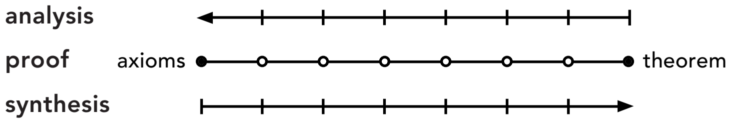


Figure 1.5 The relation between analysis, synthesis, and the proof. Compare with figure 1.8.

There is ample evidence that the method of analysis and synthesis was known even long before Pappus (Hintikka & Remes 1974, pp. 7, 85, 100). First, Pappus here refers to them as originating in Euclid and others. Secondly, there are other writers who refer to what appears to be the same things. Generally, the methods are held to have been known by Aristotle at least, and possibly invented by Plato; in these cases, we are back some 700 years earlier, in the fourth century BC. Pappus is accredited with giving the first comprehensive *description* of this as a method; like a methodologist describing a method already used by "practitioners", and for others to follow. This thereby comes very close to design methodology.

Polya

A much more recent commentary on Pappus is found in Polya's *How To Solve It* (1945), which devotes one section to a detailed account of

Pappus' method, including examples, further explanation, and drawing out certain consequences. Polya can also be regarded as a methodologist of mathematical problem solving, and he has served as the historical link from Pappus to modern-day methodology. First, he gives a "non-mathematical illustration" (p. 145):

> A primitive man wishes to cross a creek, but he cannot do so in the usual way because the water has risen overnight. Thus, the crossing becomes the object of a problem; "crossing the creek" is the *x* of this primitive problem. The man may recall that he has crossed some other creek by walking along a fallen tree. He looks around for a suitable fallen tree ... He cannot find any suitable tree but there are plenty of trees standing along the creek: he wishes that one of them would fall. Could he make a tree fall across the creek?

This example makes a very good illustration of Pappus' method: the analysis makes a chain, from crossing the creek to walking on a fallen tree, to finding a suitable tree, to felling a tree on the bank of the river, etc. The synthesis carries out what the analysis has thought out, in the reverse order. Polya elaborates:

> This succession of ideas should be called analysis if we accept Pappus' terminology. ... What will be the synthesis? Translation of ideas into actions. The finishing act of the synthesis is walking along a tree across the creek.
>
> The same objects fill the analysis and the synthesis; they exercise the mind of the man in the analysis and his muscles in the synthesis; The analysis consists in thoughts, the synthesis in acts. There is another difference; the order is reversed. Walking across the creek is the first desire from which the analysis starts and it is the last act with which the synthesis ends. ...
>
> Analysis comes naturally first, synthesis afterwards. Analysis is invention, synthesis execution; *analysis is devising a plan, synthesis carrying through the plan.* (pp. 145–146)

Most important here is how Polya explicitly spells out the relation of *planning* to analysis & synthesis, which was only left implicit by Pappus. Polya himself writes that this passage "hints a little more distinctly than the original at the natural connection between analysis and synthesis". The final quoted sentence spells it out: analysis is making a plan, and synthesis is executing it.

He also describes the order between analysis and synthesis as "natural". Elsewhere gives further comment: "it is generaly useless to carry out details without having seen the big connection, or having made a sort of *plan*. ... It is foolish to answer a question you do not understand." (p. 6)

The extent of Pappus' influence on Polya is the most striking in other parts of the book, where Pappus is not mentioned. For example, the idea of *heuristics* is central to Polya, but this concept too is usually attributed to Pappus. But more importantly, the influence is also quite clear in the parts that could be considered as the core of Polya's own contribution. The best example is in the very opening of the book, where he presents his general problem-solving schema, consisting of four parts (pp. xvi–xvii):

- Understanding the problem
- Devising a plan
  Find the connection between the data and the unknown.
  You should obtain eventually a *plan* of the solution.
- Carrying out the plan
- Looking back

This is Polya's overarching schema; it is a "method" of problem solving in every sense of the word. It is a mathematical equivalent to design methods. Furthermore, it is plain to see that this is an extended version of Pappus' original two-part scheme. A final stage of evaluation has been added for pedagogical purposes, and the analysis–synthesis schema has been refined with Polya's clarification regarding planning: The analysis has been elaborated into understanding the problem and devising a plan, and synthesis is called "carrying out the plan" (figure 1.6 overleaf).

With these ideas being so similar, is it likely that the design methodologists knew about these things; was there a link? Yes, it seems so. Although neither Alexander nor Jones gives any indication in that direction, there is good reason to believe that both were at least familiar with Polya's work. Given this suspicion, stemming from the unmistakable similarities between their methods and Polya's, one finds that Alexander (1964) cites later works by Polya, and also (Miller, Galanter & Pribram 1960) where Polya's method is presented in some detail. Jones' link is more indirect; however, the structure of (Jones 1970) is so similar to Polya (1945) that it could have been named "How

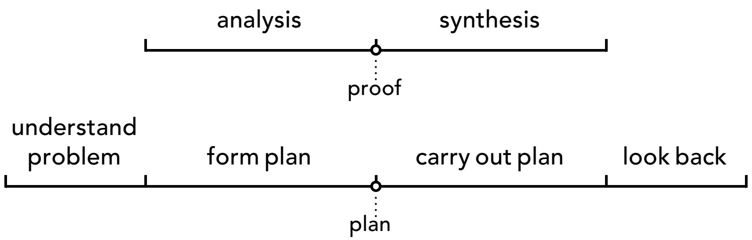


Figure 1.6 Polya's problem solving schema as an elaboration of Pappus' version. Here the proof and the plan are marked as entities passed on between the first and second halfs in both models.

To Design It": besides being based on the analysis– synthesis–evaluation pattern, both texts contain a catalog of heuristic techniques.

Descartes

Even though both of the principles of planning and product–process symmetry can indirectly be found already in Pappus, there is one point on which present-day design methods differ from the Greek original, and this is the technique of *hierarchical decomposition*. Its invention should probably be attributed to René Descartes, as part of the method he developed, a derivation of the Pappan original. This method was presented in his *Rules for the Direction of the Mind* and *Discourse on Method* (1628, 1637):

> We shall comply with it exactly, if we *resolve* involved and obscure data step by step *into those which are simpler*, and then *starting from* the intuition of those which are *simplest*, endeavour to *ascend* to the knowledge of all the others doing so by corresponding steps [taken in reverse order]. (*Rules*, rule V, my italics)

Loosely, hierarchical decomposition consists in the strategy of divide and conquer; in making a complex matter manageable by breaking it down into successively smaller parts, which taken together form a tree-structure of the problem. This is where the concepts of "top down" and "bottom up" come from. Here, analysis is the process which breaks the problem down into parts and creates the hierarchy, and synthesis is where the pieces are reassembled with the same structure to make up the solution (figure 1.7).

Like these chains, the hierarchical analysis and synthesis processes are also symmetrical to each other: they follow the same tree structure, only in opposite orders. Furthermore, they adhere to the principle of product–process symmetry: their tree structure is the same as the solution structure. This is a more powerful organizational principle than the strictly linear one that Pappus used, yet it can be regarded as refining rather than replacing the original.

Descartes developed his method with the objective of using it in his own effort to establish a systematic and absolutely certain foundation for science and all knowledge in general; the method would serve as the basis and rationale for this system.

> The aim of [the method] should be that of so guiding our mental powers that they are made capable of passing sound and true judgments *on all that presents itself to us*. (Rule I)

It was in this system that "cogito ergo sum" was made into the most certain and fundamental fact of all, and from which all other facts were to be systematically deduced.

As a system of this kind would be somewhat more complex than a geometrical proof, it called for a more powerful organizing scheme than Pappus had needed; this was presumably the reason why he developed the principle of hierarchical decomposition, something that hadn't been called for in geometry.

Descartes' definition of method

Descartes clearly stated mathematics as the source of inspiration (Rule II). Although admitting familiarity with the work of Pappus and other "ancient geometers", and stating that they had used a method of analysis, he claimed that they had hidden it from others:

> We have sufficient evidence that the ancient geometers made use of a certain "analysis" which they applied in the resolution of all

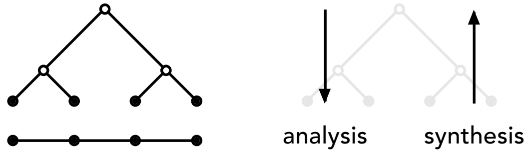


Figure 1.7 Descartes' hierarchical decomposition principle arranges the elements into a tree, whereas Pappus' version simply forms a linear chain (cf. figure 1.5).

> their problems, although, as we find, they grudged to their successors knowledge of this method.... Certain vestiges of this true mathematics I seem to find in Pappus and Diophantus... These writers, I am inclined to believe, by a certain baneful craftiness kept the secrets of this mathematics to themselves. (Rule IV)

They had withheld this knowledge, he claimed, so as not to take the luster off their own mathematical achievements—it would have made them look trivial, as nothing could be simpler than using this method (*ibid.*). He could therefore claim the method he was describing to be of his own invention. Judging from the philosophical literature on the topic, he seems to have had some success in this.

Descartes is also remembered for being the first to reason *about* methods; about their function, advantages, and so forth. Descartes also gave a definition of method:

> ... Now by method I intend to signify rules which are certain and easy and such that whosoever will observe them accurately will never assume what is false as true, or uselessly waste his mental efforts, but gradually and steadily advancing in knowledge will attain to a true understanding of all those things which lie within his powers. (IV)

Method, rationality, and logic

Another point to recognize is his claim for the method to be of use far beyond his initial aim, and not to be restricted to any specific field. He claimed it to be general-purpose, to be used in all domains with the same results, as a means for attaining "universal Wisdom":

> For this discipline claims to contain the primary rudiments of human reason, and to extend to the eliciting of truths in every field whatsoever. (IV)

In making this claim, Descartes connects method with rationality. The generally recognized concept of rationality is itself inherently circular. It is defined as the "state of reasonableness", and "rational' is defined as "of the reason", from *ratio* which means *reason*, that is, the same thing. Consequently, rational thinking thus means "thinking that is of the mind". The concept of rationality hence says nothing about just *what* reason consists of, what is reasonable, and so on, so we need to add a theory that says what reason consists of, that determines what is agreeable to reason, and so forth.

What Descartes does is to propose that *method* is the foundation of rationality. It is his method that "contains the primary rudiments of human reason", that can determine what is agreeable to reason, etc. Still today, arguably, the received view of rationality and reason is derived from method as well as logic; this holds for scientific theory as much as for lay views of rationality. Good thinking is thinking that follows a method; a particular procedure.

The best description of the relation between rationality and design methods, and methods in general, is given by Parnas & Clements:

> A perfectly rational person is one who always has a good reason for what he does. Each step taken can be shown to be the best way to get to a well defined goal. Most of us like to think of ourselves as rational professionals. However, to many observers, the usual process of designing software appears quite irrational. Programmers start without a clear statement of desired behavior and implementation constraints. They make a long sequence of design decisions with no clear statement of why they do things the way they do. Their rationale is rarely explained.
>
> Many of us are not satisfied with such a design process. That is why there is research in software design, programming methods, structured programming, and related topics. *Ideally, we would like to derive our programs from a statement of requirements in the same sense that theorems are derived from axioms in a published proof.* All of the methodologies that can be considered "top down" are the result of our desire to have a rational systematic way of designing software. (1986, p. 251, my italics)

The search for design methods is motivated by the desire for a rational design process, and the authors define rationality as "having good reason"; this is a good common-sense definition of rationality, having good reason for what you do, although the circularity is evident.

Here is also a rare mention of proofs as the ideal for design methods. This also points to logic, another important part of this picture. With method as the model for rational thinking, logic may be defined as method for thinking, and what is the discipline of logic if not the methodology of thinking and reasoning? With this in mind, and with the state of utter refinement that modern formal logic has reached, it is not hard to see why Parnas & Clements regard formal proofs as the model for rational design methods. And the history of logic follows a path that is similar to those of method and rationality, all the way back to Pappus and geometry.

This can also be seen in the format of modern formal proofs. The principal difference from the "classical pattern" described by Pappus is that the steps of the proof are only given once. From having in Pappus' days been presented first in a conceived order of discovery, then followed by the proof per se; today the first sequence is omitted, and only the proof proper is given. Hence, this "modern pattern" is equal to the second half of the classical pattern. As a consequence, a proof today begins with the axioms and from them goes through the steps that end with the proven theorem.

And as this is also the format used in mathematical proofs today, this modern format has taken over the role that the classical format (as described by Pappus) was once invented for, in mathematical proofs. Hence, modern formal logic is also directly connected to the ancient geometry proofs. With respect to its application in mathematics, this seems not inappropriate at all; more so, however, when logic is held as some kind of model for human thinking, which is not so uncommon.

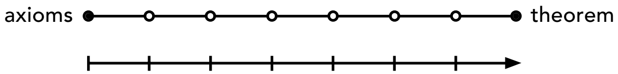


Figure 1.8 The structure of a modern proof. Cf. figure 1.5.

The domains of rationality, method, and logic are therefore tied to each other, both through their common historical origins and via principles and ideas that they share still today: Therefore (formal) logic, as the scientific study of rational thinking, may be seen as the purest form of methodology; and the principles of logic as the ideal for all other methods to aspire to, in design or otherwise.

1.3 Folk conceptions of thinking

With these ties in mind, it should come as no surprise that traces of the model of rationality can be found also in theories of cognition. More surprising, perhaps, is that the influence has been monumental, and continues to be so. But if we only looked for this influence in scientific approaches to cognition, even in the widest possible sense, then some of the most important aspects of its influence would elude us. The defining characteristics of this pattern can be found already in conceptions of the mental realm that lie well outside the domain of science. Our everyday conceptions of mind are so entrenched in these basic notions that it is hard to envisage how they might be different. An alternative conception that is not based on the classical model of rationality is hard to even imagine—remember how the definitions of what reasoning and rationality *is* are largely made in its terms, in both lay and scientific language.

The folk model of cognition

The most basic of all conceptions of the mind and its workings is the distinction made between *perception, thinking,* and *action*. This is a good example of a notion that is hard to rethink or disregard: How could it be different; how might it not be this way? Perception refers to the functioning of the five senses; how could these and action *not* be separate from thinking? Hence, this blessed trinity is said to be very deeply rooted in "folk psychology"; a term that cognitive scientists use for the body of everyday, non-scientific psychological conceptions held by every one of us, independent of schooling.

An additional aspect of this trinity is its being arranged in a linear order from perception to thinking to action, based on an imagined "flow" from input to output: Information enters through the senses and via perception goes into the mind. Then, a decision is made which transforms this information via the motor system into action, and this is regarded as "output". Taken together, this yields a folk psychological three-stage model of perception, cognition, and action (cf. figure 1.9).

Figure 1.9 The folk-psychological schema of the linear relation between perception, thinking, and action.

A second core theme of folk psychology, only slightly less fundamental than the previous one, is the *intention–plan–action* triad (cf. figure 1.10). Behind it lies the intuition that thought controls our actions and behavior in general (springing from "free will", or the like). It is essentially a causal explanation of action and how action comes about, what controls it, etc. It consists of a three-part chain where the intention plays the role as cause, which via the mechanism of planning determines action. This is typically regarded as a folk-psychological "theory" of action, or rather of the relation of thought to action: a theory since it has the characteristics of a scientific theory; psychological since it is held to explain how people conceive of thinking, action and the relation between the two; and "folk" because of its simplicity and thereby poor standard as a theory.

The primary element in this explanatory schema is planning, since that is the mechanism that performs the translation from thought to action. It takes an intention as input, specifying what the individual wants to happen, and derives from it a plan that specifies a set of operations to perform. The plan is then passed on to the motor system which executes it, i.e. carries out the specified operations, and thereby brings about the intended outcome. As the origin of the intention is non-essential to the relation between thought and action, it can be left out of this schema.

Figure 1.10 The folk-psychological intention–plan–action schema.

There is a distinct relation between the two presented schemas and the concepts therein. The first schema describes cognitive functioning on the highest and most general level of description. The second one is more specific, it concerns the relation between thinking and action, and treats a part of the general schema in greater detail. This elaboration consists firstly in introducing plans as the construct that connects thinking to action, and secondly in specifying thinking, the central part of the first schema, in greater detail: In order to realize the connection between thinking and action, a part of thinking must produce the plan that is passed on to action. The function doing this is of course planning, and it requires an intention as input. As the origin of the intention is non-essential to the relation between thought and action, it can be left out. Hence, the second schema is a refinement of the first (figure 1.11).

Connection to Pappus

How is this folk schema of cognition related to Pappus' method of analysis and synthesis? This is made clear in Polya's elaboration on Pappus, especially in the primitive-man example, which transfers Pappus' schema to a non-mathematical domain. Polya's point is that analysis corresponds to thinking, and synthesis to action.

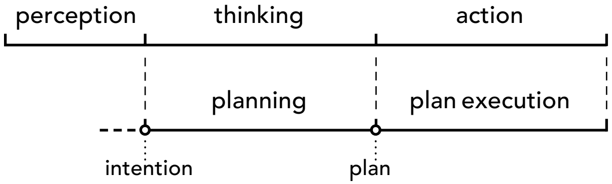


Figure 1.11 Timeline renditions of the folk psychological of perception–thinking–action and intention–plan–action, also showing how the second is an elaboration of the first.

First, the chain of steps that makes up analysis, going from the goal backward, is the train of thought that the man goes through in figuring out how to cross the river:

> The man might remember himself having crossed another river by walking on a fallen tree. He therefore looks around for a suitable fallen tree.... He doesn't find one but there are plenty of trees standing on the shore. He wishes one of them to fall. Could he himself make one fall across the river? ... *This succession of ideas should be called analysis* if we accept Pappus' terminology.

Secondly, synthesis consists in physically carrying out what thinking has conceived during analysis:

> What will be the synthesis? Translation of ideas into actions. The finishing act of the synthesis is walking along a tree across the creek. (Polya 1945, p. 145)

When Polya also relates them to planning and combines them into a whole, the relation between the cognitive schemas and Pappus' method becomes quite clear:

> The same objects fill the analysis and the synthesis; they exercise the mind of the man in the analysis and his muscles in the synthesis ... Analysis comes naturally first, synthesis afterwards. Analysis is invention, synthesis execution; *Analysis is devising a plan, synthesis carrying through the plan.* (pp. 145–146)

Here, it is quite appropriate to equate the intention with the "goal" or "the thing sought", i.e. the theorem to be proven. Planning consists of analysis, which takes the intention as its starting point. Similarly, the intention is realized when the final step of the synthesis is performed. The result is a one-to-one match with Pappus' method,

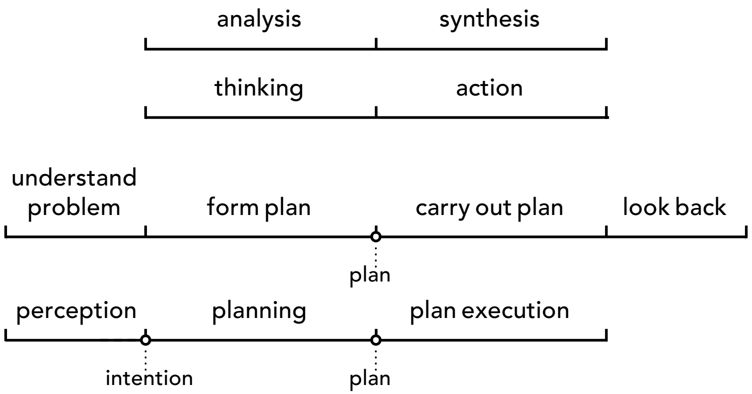


Figure 1.12 Comparisons between the basic and elaborated versions of folk psychology and Pappus' (Polya's) method.

between both the basic and elaborated versions, where the latter in both domains are yielded by adding planning (figure 1.12). This folk schema of cognition embodies the idea of intramental cognition. This view, which is entirely consistent with our intuitions, hinges on the separation of action and thinking, among other things.

Aristotle and the origin of folk psychology

So folk psychology follows very closely the ancient model of mathematical rationality. Or is it the other way around? Is this model based on everyday conceptions? There is a passage by Aristotle on deliberation (*bouleusis*) or "planning" compared to analysis, where he derives the equivalent of the intention–plan–action schema from the method of geometrical analysis (cf. Hintikka & Remes 1974, pp. 85f); it supplies us with evidence that contemporary folk psychology too is based on the ancient Greek method of analysis.

First it should be established that this account of "deliberation" really is concerned with planning, and not careful thinking in general as the term signifies today. Aristotle states that an individual's deliberation is concerned only with what she can do, what her own efforts can achieve. That is, with *action* (Aristotle, *Nichomachean Ethics* III:3, my italics):

> But we do not deliberate even about all human affairs; for instance, no Spartan deliberates about the best constitution for the Scythians. *For none of these things can be brought about by our own efforts. … We deliberate about things that are in our power and can be done*; and these are in fact what is left. For nature, necessity, and chance are thought to be causes, and also reason and everything that depends on man. Now every class of men *deliberates about the things that can be done by their own efforts*. And in the case of exact and self-contained sciences there is no deliberation, e.g. about the letters of the alphabet (for we have no doubt how they should be written); *but the things that are brought about by our own efforts, but not always in the same way, are the things about which we deliberate*, e.g. questions of medical treatment or of money-making.

Hence, deliberation is concerned with things we can make happen, but for which it is not obvious how to attain them; deliberation serves to figure out how to do this. It is therefore planning of action that Aristotle is discussing here.

Further, he states that ends, i.e. goals, purposes, aims, are not at issue in deliberation but are regarded as given. That is, the agent's intentions are not at stake, only how to attain them. This closely matches the view of planning as the mapping from intention to action:

> *We deliberate not about ends but about means*. For a doctor does not deliberate whether he shall heal, nor an orator whether he shall persuade, nor a statesman whether he shall produce law and order, nor does any one else deliberate about his end. *They assume the end and consider how and by what means it is to be attained*; and if it seems to be produced by several means they consider by which it is most easily and best produced, while if it is achieved by one only they consider how it will be achieved by this and by what means this will be achieved, *till they come to the first cause, which in the order of discovery is last. For the person who deliberates seems to investigate and analyse in the way described as though he were analysing a geometrical construction* (not all investigation appears to be deliberation—for instance mathematical investigations—but all deliberation is investigation), *and what is last in the order of analysis seems to be first in the order of becoming*.

Aristotle here provides conclusive evidence that his account of planning is directly based on the pattern of geometrical proofs. He even explicitly names geometrical analysis as the archetype for it, but even regardless of this mention the evidence is unequivocal: There is the assuming of the sought end, and then considering how it is reached; then there is the repetition of this "till they come to the first cause", which in turn is done in the reverse order in the "becoming" where the actions are carried out. Note that deliberation is compared to analysis only; not synthesis. Also, from the manner in which the final clause describes "becoming", it can only be understood as though action is held to follow planning in the same manner that synthesis follows analysis (not only in the temporal sense).

A closer analysis of the connection between Pappus' and Aristotle's accounts is made in Hintikka & Remes (1974, chs. 1 & 8), showing also extensive terminological similarities. However, they still hold that Pappus did not directly draw on the work by Aristotle.

The everyday meaning of "design"

With this in hand, one can analyze the conventional, everyday meaning of the term "design" itself. It is used both as noun and verb, where the verb, as in the process of design, is what the present book is concerned with. Keeping in mind that etymology is always a precarious venture, "design" comes from the Latin *designare*, to designate, which here means to *specify*, as in pointing out what to do. The modern sense of design is held to have originated in the Renaissance, when architect and builder functions came to be two separate functions. The architect would no longer always be present on site during building and therefore had to specify what to build, which previously hadn't been necessary (Herbert 1993).

Similarly, the noun "design" comes from *signum*, which is *not* so much in the modern sense of the root "sign" (as in symbol, mark; semantics, semiotics, etc.) as is sometimes claimed. It rather has the meaning of something that you follow, in the sense of the specifications passed on from architect to builder.

> Around the sixteenth century, there emerged in most of the European languages the term "design" or its equivalent. The emergence of the word coincided with the need to describe the occupation of designing. ... Above all, the term indicated that designing was to be separated from doing.
>
> (Cooley 1988, p. 197, quoted in Bødker et al. 1991)

In addition to this standard meaning of the noun, there are some peculiar senses which give indications toward the folk conception of how designing is done (definitions and etymology taken from the *American Heritage Dictionary*, synonyms from Roget, my italics):

> **de·sign** *n.* ... 7. A *plan*; a project. 8a. A *reasoned purpose*; an *intent*. b. *Deliberate intention.*

Here one somewhat unexpectedly finds evidence of the everyday view of design activity, given in the very definition of design. The given meanings include plan and intention, which both are part of the intention–plan–action schema. These folk conceptions are clearly expressed in the definition of the verb:

> **de·sign** *v.* 1a. *To conceive or fashion in the mind*; invent: design a good excuse for not attending the conference. b. To *formulate a plan* for; devise: designed a marketing strategy for the new product. 2. To plan out in systematic, usually graphic form: design a building; design a computer program. 3. To create or contrive for a particular purpose or effect: a game designed to appeal to all ages. 4. To have as a goal or purpose; *intend*. 5. To create or execute in an artistic or highly skilled manner.

The present topic is accurately captured by the sense listed as second—the illustrations are those of the dictionary—and sense number 5 is a variant thereof. The senses of the verb that remain capture the folk psychological view of design—" folk design theory" if you will: 1a states that design is an act of the mind; 1b (and 2) refers to planning, and number four mentions intent. Hence two items in the intention–plan–action triad are mentioned. When this connection is stated in the standard definitions, it strongly backs up a deeply rooted connection between the meanings of plan, intention, and design. The logic behind this becomes clear if these folk psychological terms are seen in relation to Pappus' schema (figure 1.13 below).

Here, action is the final step, whereas intention and plan both belong to the domain of thinking. Hence, in this view, design corresponds to transforming the goal into a plan for the implementation; a blueprint for how the product is to be executed. In other words, design corresponds to planning, or more roundly, to thinking. By extension, the design–implementation dichotomy corresponds to those of planning–execution, thought–action, and analysis–synthesis; a deceptively natural fit that surely has a large share in the appeal of this view. Note that this separation of design and implementation corresponds to the original division in Renaissance architecture.

Points 3 and 4 in the definition bear witness to the role of the intention in this scheme. At first sight they seem indistinguishable;

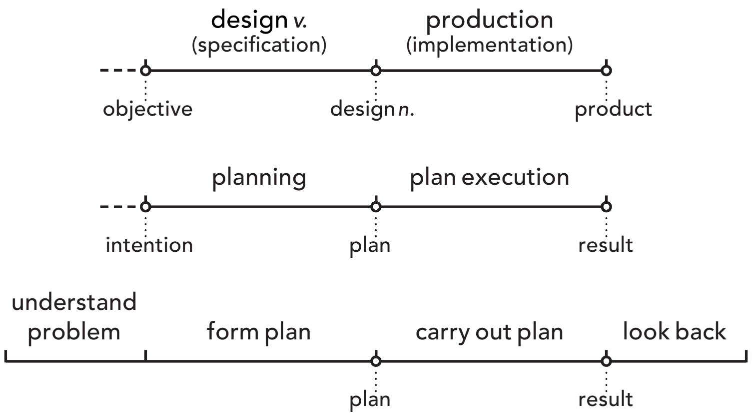


Figure 1.13 The folk-psychological notion of design, cf. the intention–plan–action schema (middle) and Pappus'/Polya's method (bottom).

however, the first refers to *creating* for a certain purpose; the second to *having as* purpose. In other words, having an aim vs. turning it into to a plan for its implementation. Folk psychology holds goals and purposes to always have an explicit mental entity, the intention, that directly corresponds to it. The intention is the cause that alone is responsible for starting the process of design, as well as determining its direction. The relation between intention, plan and design is verified by the definitions of the other two terms; even though being synonymous is not the same thing as being identical, this shows that all three terms are given their meaning in relation to the same intention-to-action schema:

> **in·tend** *v. tr.* 1. To have in mind; *plan*. 2a. To *design* for a specific purpose. b. To have in mind for a particular use. 3. To signify or mean. *intr.* To have a *design* or purpose in mind.
>
> **plan** *v.* 1. To formulate a scheme or program for the accomplishment, enactment, or attainment of. 2. To have as a specific aim or purpose; intend. 3. To draw or make a graphic representation of.

What is more, intention, plan and design are all three consistently listed as *synonyms*, even, and this in the relevant senses:

> To have in mind as a goal or purpose: *intend*, aim, contemplate, *design*, envisage, envision, foresee, *plan* ...
>
> To form a plan/strategy for: *design*, *plan*, *think out*, prepare, outline, scheme, lay plans ...
>
> What one intends to do or achieve: purpose, aim, *design*, goal, intent, *intention*, meaning, *plan* ...

So design and implementation eventually came to be assigned to separate parts of an individual's faculties, thought and action. In the very beginning, both design and building were done by one person, requiring her full physical and mental capacities. When design came to be considered as an activity of its own, it referred to one out of two responsibilities divided among two people, each of which still required one individual's full capacity.

In the folk sense it is implicitly assumed that one's thinking alone is sufficient for designing. This is the same effect that the Greek schema has had on our received conceptions of cognitive abilities in general: one's intellectual capacity is held by the mind alone, and intelligent performance is done strictly by thinking, such as in reasoning, problem solving, and so forth.

As the folk sense of design refers to the act of specification, which is only one aspect of a whole design process, I will use *specify* to refer to that particular aspect, in a theory-neutral manner. In relation to this, the topic of this book could be design as in *the whole process of design*.

How "folk" is folk psychology?

The passage from Aristotle shows that what today is considered as the folk-psychological model of action was in fact once upon a time derived directly from the method of geometrical analysis. In the same manner, "folk design theory" seems far too refined and consistent to be a reflection of mere intuitions. This casts some doubts on the view of folk psychology as a direct reflection of everyone's "intuitive" conceptions, which is customary in cognitive science.

This view is problematic by itself. Folk psychology is commonly thought of as a "folk" theory of psychology, as opposed to scientific psychological theory. That is, people are held to have "theories"; there are other examples, such as the "theories of mind" thought to be held by infants as well as primates, or "naive physics". However, it is questionable whether people's conceptions of these matters can be regarded as theories, and one may wonder whether scientists are not too self-indulgent when they attribute these tools of their own professional trade to whomever they study. As "folk" conceptions typically are neither consistent and coherent, nor well integrated and complete, the question is whether calling them theories is not going too far.

It is my belief that when scientists assemble people's conceptions and from them formulate a "folk theory", a coherent whole, they are in doing so attributing to these people a consistency, completeness, and so forth which only has come to exist as a product of the scientists' own research effort. The coherence required for this to be a theory is then something their own labor has had to produce, it is not a property of the material from which they built it.

Therefore, what is called folk psychology is not so very "folk", but rather a semi-scientific theory that is based on or derived from lay conceptions, although far from being a pure reflection of them. These intuitions are in themselves not at all as articulate, coherent, or consistent as this product, the "folk" theory. I believe the same is true for dictionary definitions. What person on the street would you expect to give a coherent and complete, eight items long definition of the noun "design"? Dictionary editors, however, typically do have scientific training.

What is more, these lay conceptions are often described as "intuitions", and the intuitions which the scientist used were typically her own. How representative can we consider these to be? How untainted by their professional experience and education? No ordinary folks would consider scientists to be ordinary folks. It often seems the veil of folk psychology is used to cover the fact that the ideas scientists are describing or defending are in fact their own intuitions, a fact which they do not quite owe up to (e.g. Fodor 1975).

Also, science and everyday life are not perfectly isolated domains. They continually influence each other through the culture they share. For instance, Freudian theory has had a tremendous impact on popular culture (an impact which remains strong, while its weight in science continues to fade). More contemporary are the information and computer metaphors, the impacts of which are beginning to follow the same pattern. Similar patterns are easy to imagine, going at least as far back as into antiquity as in the examples we have just seen, and the passing of ages tends to render them invisible.

All in all, folk psychology is arguably not so "folk" as is sometimes thought. More likely, models of this kind are highly refined derivatives from everyday conceptions, selectively chosen and subjected to scientific method to yield what resembles a coherent, plausible theory. Hence, the "folk" models of thought and action, of design, and so on that we have seen here, have therefore had plenty of opportunities for influence from modern science or ancient writings. In other words, in the good two thousand years that have passed, there has been plenty of time for the original Greek ideas to infiltrate contemporary semi-formalized accounts of these not so mundane topics. And given the *de facto* ties of these subjects with science and academia, it is not as unlikely that they retain a strong influence on even the most refined modern theories of design, problem solving, cognition, etc.

1.4 Cognitive planning theory

Plans and planning have gradually emerged as central concepts: Polya shows that the plan is what holds analysis and synthesis together, he holds analysis to consist in producing a plan, and the proof to be the same thing as the plan; Aristotle took geometrical analysis as the paradigm for planning in human thought; folk design theory holds design to correspond the planning stage in the folk model of cognition, and the "design" is regarded as a plan for building the product; the plan is regarded as the construct that connects thought to action; the third principle concerns planning, etc.

Why does planning keep recurring in this way? The reason why it turns out to be so central is that it is the mechanism by which thinking determines action. In the received, "folk" model of cognition, rational action is an extension to the model for rational thought. There is a model of rational thought, based on geometrical analysis, mathematical proofs, etc. *Acting* rationally means having good reasons for your actions; it is simply explained as a direct extension of rational thought: first you think rationally, then you act out your decision. This also follows the intuition that our minds control our actions.

In this scheme, planning is the construct that realizes this connection: as the last part of thinking, it produces a plan, which is passed on to be executed by the motor system.

Hence, planning theory and the intention–plan–action schema are in fact the received theory of (rational) action. It is based on the following principles: Given, to begin with, is the rationality of thought; add to this the separation of thought and action, and that thought precedes and determines action. Planning and the plan are required to connect the two that are separated.

From this it follows that planning is not a freely elected theoretical choice from a number of available alternatives. With thought and action being separated, which is done already in the basic perception–thinking–action schema, and free will or thought determining action, there is little option but first to place thought prior to action, secondly to introduce an entity needed to connect the separated parts, and thirdly to require the first separated part to produce this entity (i.e. to introduce a planning process into thinking).

In this way, planning theory comes to represent this relation in principle between thought and action, it becomes the concrete manifestation of this abstract principle. This is also why planning keeps coming up in so many different contexts.

What is more, since the planning concept has been directly and completely adopted by cognitive science, it has also become a cornerstone of the standard scientific theories, directly reifying the intuitive notion of thought or free will determining our actions. Planning theory has thereby become the standard scientific theory of action.

The first work to explicitly propose planning as the mechanism connecting intramental thought and action in cognitive theory was *Plans and the Structure of Behavior* (Miller *et al.* 1960). Their chapter on planning in problem solving also draws directly on the work of Polya (1945), with even a hint of its ancient origins. Hence, in this case the link from Pappus' method of analysis via Polya to cognitive planning theory was much more direct and explicit than via folk psychology only.

This is certainly why there has been such a great controversy over planning in cognitive science (e.g. Agre & Chapman 1990, Brooks 1990, 1991a, 1991b, Suchman 1987): this has not been a dispute over people's grocery shopping lists, or how they go about deciding how to spend their summer holidays. The row over planning has concerned the essential nature of human action; needless to say a fundamental issue also for cognitive science. What is more, such a great part of the work that has been done in this discipline hinges on this particular theory of action. If it doesn't hold, the work based on it stands under serious question. For this reason, planning is not a harmless, isolated little sub-aspect of interest only to a "planning community" of researchers, and the enormous heat that has been generated by this debate is understandable. This would not have happened over a

theory of shopping lists. What is interesting is that those explicitly concerned with planning research mainly seem to have taken the consequence of this critique, and have abandoned the received view to create a new field called "reactive planning". This remains to happen outside these circles.

Design as planning

In a paper that is arguably a classic, Jeffries *et al.* (1981) proposed a theory of design as planning, directly based on cognitive planning theory. The authors themselves however do not regard this as a *planning* theory of design as such, just as a cognitive theory. This also goes to show just how fundamental the concept of planning is to cognitive science.

This paper is also significant for another reason: the main method for studying design empirically is to collect protocols of designers thinking aloud while working on a design problem. This is also probably the first protocol study of design. It is also the first of three protocol studies of software design, the other two of which will be treated in the following chapters.

The paper ties together work from three disciplines: The first is design methodology, especially the part concerned with software design. The second is "automatic programming", i.e. attempts at design-by-computer, a branch of AI, and the third is cognitive research on "models of planning and design" (as notably the authors themselves call it). The last two represent the classical AI–cognitive psychology connection in cognitive science. How compatible all three disciplines are is shown by how smoothly their contributions are combined.

Already in the first paragraph, the description of the nature of design is entirely permeated with concepts from planning theory:

> The task of design involves a complex set of processes. Starting from a global statement of a problem, a designer must develop a precise plan for a solution that will be realized in some concrete way (e.g., as a building or as a computer program).

Here as often elsewhere, what is taken for granted in the paper is particularly evident in the very opening statements. Even when the authors here give their introductory description of what design is, they do so in terms of planning theory. This is not an analogy; it is not design *as* planning—design *is* planning. Design consists in developing a plan for the implementation, by translating the given goal into a specification of what should be done:

> Software design is the process of translating a set of task requirements (functional specifications) into a structured description of a computer program that will perform the task.

This definition completely parallels how the dictionary above defined design in terms of folk psychology and planning (cf. figure 1.13). The resulting design consists of a decomposition of the product into modules; the plan for the implementation process follows this decomposition structure:

> One can think of the original goal-oriented specifications as defining the properties that the solution must have. The design identifies the modules that can satisfy these properties. How these modules are to be implemented is a programming task, which follows the design task.

1.5 Problem solving theory

Problem Solving Theory (PST) and Information Processing Theory (IPT) are quite closely related; roughly, the latter is a generalized version in which the former has been made into a universal, domain-independent theory of cognition. Both theories originated in the work of Newell & Simon. The main presentation of these theories is considered to be their book *Human Problem Solving*, which was published in 1972. This was a good fifteen years after their research had begun, so this book must be considered as a mature presentation of their work, even though it may seem old today, or limited in its empirical material. The book, and the theory of problem solving, are based on the study of three problem solving tasks: logic, chess, and cryptarithmetic (i.e. substitute digits for the letters in DONALD + GERALD = ROBERT). And although problem solving is the specific scope of the book, it is also considered to be the major scientific work on information processing theory as a whole. This if anything is a testament to the relation between these theories, and to the status assigned to problem solving as an activity that is representative of human cognitive performance altogether. That is, to the relation between problem solving and cognition as a whole, as it is perceived by cognitive scientists.

These theories bear strong evidence of their origins. The problem with which Newell & Simon begun their research was to construct a computer program capable of proving theorems in formal logic. Both their choice of problem domain as logical proofs, and of computer implementation as their method, are clearly visible in the resulting theories, even though these are formulated in general terms.

Their first work was on a program known as the Logic Theorist (LT), which later was superseded by the General Problem Solver (GPS). The task of LT was to find a proof for a given theorem in formal logic, given the axioms to be used for the proof. For a program to be able to do this, everything needed first had to be encoded appropriately and given to it. Besides axioms and theorem, also the available rules of logic, their proper application, and how they are combined into deductive sequences, had to be encoded into the program in an appropriate form. Then, using various methods, the program was to assemble valid combinations of steps into a sequence leading from axioms to proof.

In order to turn this from a computer program into a theory, the various aspects of the program were expressed in mathematical form. The process was characterized as a search among the available rules, and their possible applications and combinations. This given information that had been fed into the computer was collected in what was called a "search space", to match the analogy of the program's function as a search. This space is a mathematical abstraction with no immediate counterpart in the computer program, but embracing various parts of it.

According to historical sources, Newell & Simon drew directly on the work by Polya (1945) in their initial work on LT, and followed his directions closely in developing their first algorithms. The most direct indication of this connection lies in that LT worked strictly backward from theorem to axioms in its search for proofs (Newell & Simon 1972). Additionally, given their oft mentioned concern with heuristics, they could hardly have missed Polya since it is his name that is most often associated with the study of heuristics in modern times. The one missing piece of evidence is that they themselves seem not to acknowledge this connection. (Still it would be unfair to imply that they were attempting to do a Descartes in this matter and deny the origins.) But using Polya's recognized work on techniques for mathematical problem solving would of course be a wise thing to do, in attempting to develop a computer program with this ability.

When they extended their work to other tasks than logic, when the

Logic Theorist became the General Problem Solver and they formulated a universal theory of problem solving, the previous concepts were simply generalized: The initially given axioms and theorem became start and goal states, defined by the problem in question; the basic "search" view was transfered to problem solving, now taking place in a "problem space". The rules of logic were replaced by the rules of the particular problem, and so on. Hence, the basic pattern from logical deduction was preserved and became the backbone of the theory of problem solving.

At this point Newell & Simon were aiming toward greater psychological realism, and started to collect their own think-aloud protocols of actual subjects working on the problems, with the purpose of replicating such protocols in their programs. With this aim, and from the analysis of these protocols, they developed a new heuristic method called "means–ends analysis", to emulate in GPS what their subjects were doing (1972). Means–ends analysis allows for working alternatingly forward and backward, and it is therefore a more powerful method than working strictly backward (or forward). In conjunction with this, they also developed a heuristic called the "planning method". It consists in developing a simplified solution that abstracts from the specifics of the problem, and thereby can establish a general solution strategy, which then works as a plan that is used for the "implementation" of a solution that does deal with the details.

The first thing to note about these various "heuristic methods" is that they all correspond to the analysis part of Pappus' original schema. The original backward-working method of the Logic Theorist even exactly corresponds to Pappus' method of analysis for a proof problem: both start at the theorem to be proven, and move backward until they reach the axioms, and at this point the resulting steps are delivered as the proof. The only difference is that LT requires that also the axioms be given in advance. Means–ends analysis replaces that strategy with a generalized and more powerful method for finding the proof, but it *only* changes the method for finding it—the internal workings of the analysis stage—the givens are the same, axioms and theorem, and the result is the same, the proof.

The same holds also for the planning method; it changes only the method of analysis while preserving everything else. What is more, this method of analysis is identical to top-down decomposition in cognitive planning theory: it organizes the analysis phase in the same way that planning theory does. There are also great similarities between the planning method in problem-solving theory and Polya's "creating a plan". So these may be seen as three different "decomposition strategies", cf. Descartes' hierarchical principle.

The influence of folk psychology on current cognitive theory

But the most important point in this is that these are three variants of the analysis stage in Pappus (counting in strict backward movement). They are surely not equally powerful, but neither of them goes beyond analysis, which corresponds to thought in the folk-psychological schema, nor do they deviate from the rest of the schema on any point—the division into perception, thought, and action is preserved, as is planning as the mechanism connection thinking and action. And since the goal is considered as given, also the intention is left as is. They differ in the inner workings of the thinking box in the middle, but the rest is left intact as it has been in folk psychology all along and still is; they are different theories of thought, and of how thinking produces the plan for action (figure 1.14).

And this is what is striking about the current state of cognitive theory: while vast amounts of work have been spent on the details of the inner workings of the mind, the folk schema of cognition has been adopted by scientific theory on all other points, and stands unaltered even today. This holds for both planning theory and problem-solving/information-processing theory. The fundamental organization of the cognitive system has thereby been preserved more or less unchanged from Aristotle's treatise on planning. That is also why this section, even though serving to present cognitive theory, is dominat-

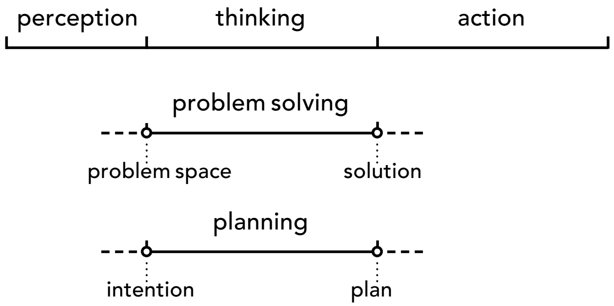


Figure 1.14 Problem-solving theory and planning theory as variants of the "thinking" step in the folk-psychological schema.

ed by the presentation of the pre-scientific, folk conceptions of cognition: These notions are where the majority of the concepts originate that are relevant for these theories. The continuous, unbroken line of development from folk psychology to modern-day cognitive science is especially clear in the case of planning and planning theory.

From this it is clear that today's cognitive theories also preserve the intramental view of cognition. One might say that these theories disagree over the intramental details, while they all preserve the basic patterns from folk psychology that are derived from Pappus, Aristotle, and so forth. When connectionism was up and coming, it was widely hoped that it would replace traditional information-processing theory, or what was then seen as the symbolic theories. It was seen as a question of symbolic vs. connectionist theory. It has since become clear that this replacement would not happen; the connectionists never really took on the higher-level issues. (Although Rumelhart *et al.* 1986 hinted at how this might be done.) Instead they took the route of the natural sciences, down into the biology, even the chemistry. That is why information processing theory and the work of Newell & Simon still stands as the best alternative; more or less unchallenged, even.

The reason why I will use "intramental" to refer to the traditional theories is also to a large extent just this: The fact that these are symbolic or information-processing theories, etc., is beside the point. What is significant is that they retain the view of cognition as intramental—a fact that connectionism hardly challenges, for example. This is the aspect of traditional theory that I will point out as the problem in what follows.

I am presenting PST here for several reasons. One is that it is the most influential theory of cognitive science. It was very early, pre-1960 even, and it has served as the basis for very influential, more recent theories: SOAR (Newell 1990) as well as GOMS (Card, Moran & Newell 1983). An additional but related reason is that PST definitely is the most developed cognitive theory when it comes to design specifically. This was explicitly named by Newell & Simon (1972) as a suitable domain for taking the theory to more realistic, more complex problems. This they have also done, in particular Simon, who has shown an interest in design (e.g. 1973, 1981). It has also been done by a number of others (e.g. Akin 1986, Goel 1995).

But the main reason for discussing problem-solving theory here is that I will use it as the basis for my critique. I have made this choice because it is the *best* theory that cognitive science has to offer, with respect to the topics I will be discussing. That is to say, I have *not* chosen it as a bad example. Even though my argument will to a large extent be based on the flaws and problems with this theory, this should not be seen as much as a criticism against Newell & Simon. As scientists they are among the very best. And problem-solving theory is not a bad theory (it is wrong, however). In studying their work, even though I have not been on their side, I have come to regard them highly for what they did. For example, they did consider and work out the issues of how their theory would connect to action and the outside world, to an extent that no one else has done (Newell & Simon 1972, Simon 1981). Their theory is also coherent, and regarding these issues, they would not easily lend themselves to making *ad hoc* extensions that were not compatible with other parts of the theory, as is too often done.

Hence, the critique of problem solving theory should not be seen so much as a critique directed toward Newell and Simon, but rather toward the rest of cognitive science, which in the good 25 years since has hardly even come up with anything significantly different. The only real alternative that might be considered here, for example, is mental models theory (Johnson-Laird 1983). In all, it seems cognitive science has ever sincebeen little more than footnotes to Newell and Simon.

1.6 Summary: A common model of rational action

In the introduction I stated that this chapter would establish a connection between design methodology and cognitive theory. These are two disciplines that seem quite unrelated at first sight. Here I have shown the link to be a shared underlying pattern that is *a general model of rationality and rational action*. This model has been the foundation for such apparently diverse domains as classical design methods and structured development techniques in software engineering; with respect to cognition, folk psychology, cognitive planning theory, problem-solving theory, and information-processing theory; and lastly, to proof theory and formal logic, and the philosophical notions of method and rationality.

Although they vary in their details, for various reasons, these widely different domains all are founded on this underlying pattern. This pattern is what the four defining principles are meant to make visible, and they should be seen as my attempt to make this model of ra-

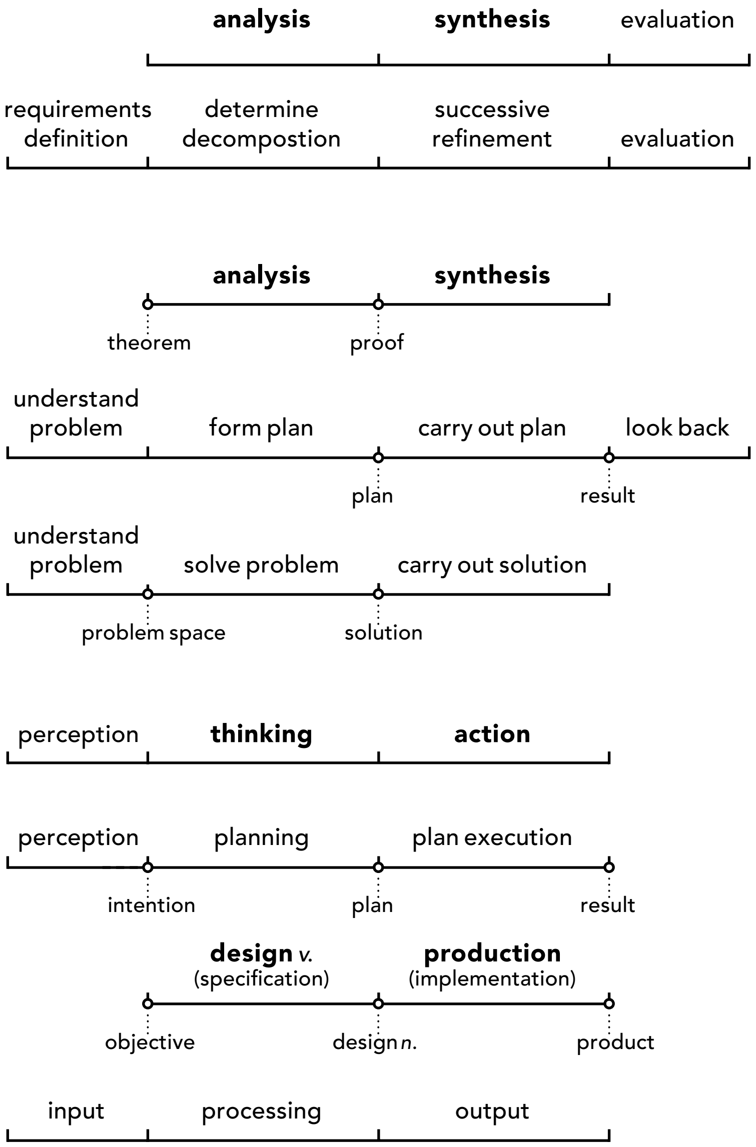


Figure 1.15 The Rosetta stone of rational action models. The models belong to three categories: design methodology, logic/problem solving, and folk psychology. The final line is the basic pattern of computer architectures.

tional action visible and explicit, and to articulate its essential underlying ideas.

By placing the various timeline diagrams beside each other, you get a Rosetta stone of rational action models, as in figure 1.15. There, the models can be compared and their corresponding elements can be read out directly. For example, the analysis–synthesis dichotomy lies behind those of thought–action, planning–plan execution, and design–implementation. These pairs make up the stem of the family tree, as it were. (The original Rosetta stone was found in the Nile delta, not far from Alexandria.)

An intramental model of rationality

This underlying model of rationality and rational action, which is based on ancient geometry proofs, stands as an archetype of good and desirable thinking, and it is the paradigm after which all the descendants have been modeled. The four principles capture the nature and essence of this archetype, showing that it is essentially an *intramental model of cognition and action*: The central idea is of rational *action* as an extension of rational *thinking*, and the elements of this idea is expressed in the four principles. Taking the rationality of thought as a premise, they make the following points, each in turn:

0. The rationality of thought.
1. *separation*: The separation of thought and action.
2. *logical order*: Thought preceding and determining action.
3. *planning*: Plans (and the intention-to-action schema) as the mechanism whereby thought pre-determines action
4. *product–process symmetry*: The idea that the *structure* of a *product* of action directly reflects structure of the *process* which produced it, and thereby also of the underlying *plan*.

In particular, the final point implies that since this structure was inherent also in the plan, it thereby had to be known before the start of the process which produced it.

As seen in the lower part of the Rosetta stone, the basic computer architecture of input–processing–output also follows the ancient pattern, with "processing" corresponding to analysis and thinking. This connection is likewise evident when Jones compares the designer to

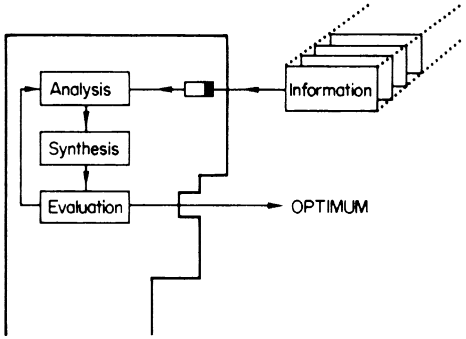


Figure 1.16 Original caption: "Designer as computer".

a computer—here also note the labels on the boxes inside the head in figure 1.16 (1970, p. 50):

> The picture of the rational, or systematic, designer is very much that of a human computer, a person who operates only on the information that is fed to him, and who follows through a planned sequence of analytical, synthetic, and evaluative steps and cycles until he recognizes the best of all possible solutions. This assumption is, of course, valid in the case of computer optimization of the variables within a known design situation, but it also underlies such systematic design methods as morphology and systems engineering which are intended for the human "computer" to use in solving unfamiliar design problems.

Here, Jones at once ties together the basic three-stage design schema, the input–processing–output schema, and thinking/problem solving. Jones also mentions "cycles"—notably in relation to computers—or *iteration* as is today's term; a ubiquitous feature of contemporary design methods; also notice the backward arrow from evaluation to analysis in the "flowchart" inside the head. The reader may have wondered why I have left this prominent aspect out of my analysis of design methods; a brief answer is that this is an *ad hoc* extension that goes quite counter to the principles of the underlying model of rationality, in effect proving it wrong; also cf. chapter 3.

2. The failure of design methods

2.1 The general failure of design methods

Having said this much about design methods, there is but one thing to add: They don't work, and they don't work at all. In spite of all the good motives—the need for potent and up-to-date design procedures, the noble cause of being rational, and so on—the failure of these methods is a very solid and widely recognized fact, as is the thoroughness of this failure.

A number of circumstances bear testimony to this. One is that it was the original advocates of these methods who documented the failure, and who then abandoned them altogether. This is a unique circumstance; usually, it takes critics from the outside to bring a failure into the light, and this is typically met with a frenetic defense from the original proponents who rarely change sides, either refusing to see or accept the failure. Here, the pioneers allowed themselves to recognize the failure of their own ideas and inventions, and to publicly state this as a fact.

A second circumstance is how exceptionally soon this reversal came. Alexander's classical description of his method was published in 1964. In 1966 he wrote an essay explaining why it didn't work. When he was interviewed about the state of the field in 1971, he dismissed design methods completely:

> And there is so little in what is called "design methods" that has anything useful to say about how to design buildings that I never even read the literature any more.
>
> ... I think I just have to be consistent here. I would say forget it, forget the whole thing. Period. Until those people who talk about design methods are actually engaged in the problem of creating buildings and actually trying to create buildings, I wouldn't give a penny for their efforts.

Also Jones (1970) acknowledged the problems with these methods, and the lack of success stories. This he did already in the original edition; even in the same paragraph as where he stated the *need* for these methods:

> However, it is not obvious that the new methods that are reviewed in this book are any better. *There is not much evidence that they have been used with success, even by their inventors* ... The usual difficulty is that of losing control of the design situation once one is committed to a systematic procedure which seems to fit the problem less and less as designing proceeds. (p. 27, my italics)

Later, Jones too would reject them completely (cf. later editions of Jones 1970). The lack of successful applications mentioned here is a third aggravating circumstance which was also generally recognized; Rittel was another pioneer who acknowledged this absence:

> Q: What kinds of problems has design methodology successfully attacked?
>
> ... If you are asking for examples from architectural design I wouldn't know of any building that has been done discernibly better than buildings done in the conventional way.
>
> ... I would say that the corporations or other planning institutions who seriously tried to accomplish something with the [design] methods have been disappointed, and that there is a considerable "hangover" from these methods. (Rittel 1972)

Also Alexander and Broadbent made similar comments:

> In short, my feeling about methodology is that there are certain mundane problems which it has solved—and I mean really incredibly mundane. (Alexander 1971)

> Yet asked to catalogue its achievements, in terms of buildings built, cities designed, and so on, most of its advocates find themselves in difficulties. (Broadbent 1979, p. 41)

Even the *attempts* to use the methods have been exceptionally rare. According to Lawson (1980), there has been only one documented attempt to use Alexander's method (Hanson 1969).

Since this failure is very well documented, I will not belabor the argument here; the existing critiques are quite sufficient. (Beyond the sources that have been cited here, see also e.g. Broadbent 1973, Lawson 1980, and several papers in Cross 1984).

2.2 The classical mistake

When it comes to understanding why the methods failed, the fundamental problems can also be traced all the way back to Pappus. The problems that are present already in his original account have since been carried over into all methods that descend from it, including the most modern ones. I will here begin by briefly examining these problems in the original context of geometry proofs.

The structure of the genuine process

The difference between analysis and synthesis does not stand out as clearly in a proof as it does in a conventional mathematical problem, the other of the two kinds of use that Pappus describes. In solving a problem, analysis consists in figuring out what calculatons are needed for reaching the answer, and synthesis of actually carrying out these calculations. As a consequence, no calculations are to be made during the analysis; this phase serves only to determine which ones *should be* performed (cf. Polya).

This division is however not made in practice, as anyone can tell from their own experience: When working on a mathematical problem, you do not refrain from attempting to calculate the answer as you are trying to find the solution. Usually, you do quite the opposite: you carry out various tentative calculations in order to figure out how to reach the solution.

You do this for several reasons: For instance, to see what you can get from what is given; you can then use this for further calculations. Or, when you have an idea for a solution, or for a part of it, to evaluate what you have and to see where it leads. Many of these calculations are dead ends; still, one or two may give you the critical insights. Why would you not make these interim calculations? There is no good reason not to, so people do make them.

Therefore, there is no phase of pure analysis, as this activity is intermingled with synthesis; nor is analysis alone sufficient to produce a solution.

As a consequence, when you have figured out how to solve the problem, then this is hardly followed by a *bona fide* synthesis phase. At that point you have already performed most or all of the necessary calculations, in particular when testing to see whether you indeed have reached the correct solution. Thus, there is hardly such thing as a synthesis phase that *a*) is performed after the solution has been found, *b*) is separate from an analysis phase, *c*) that follows after analysis, or *d*) that even exists at all.

But the work that *does* remain when you have reached the solution, especially if you are to present it to someone else, is to go back to the pieces of writing on your paper, extract the parts that go into the solution, and write them down in the logical order. *This* is how you obtain the straight path from what is given to the solution, the sequence of steps that corresponds to a proof, and that according to Pappus is followed during analysis as well as synthesis, without redundant parts and dead ends.

In actuality, this path is never followed; not during any part of the process. Hence it is a mistake to see this as plan following, or to think of the proof as a plan. It cannot even be *known* until at the very end of the process, when you also have the answer. Only then can you extract this path, and it still requires effort to obtain it: you have to look back at what you have done, go through your scribbles, and then assemble the pieces into a tidy, linear sequence.

Instead there is just one process, where the functions of analysis and synthesis are two aspects of the same activity, not two different activities, stages or processes. The structure of this joint process is anything but a clean progression from desired result to given facts, neither backward nor forward. This process will by necessity follow many routes that turn out to be dead ends, and others that are not used in the final proof. (All these things given that the proof sought or problem to be solved is not trivial; in this case the problem is so simple that it is possible to go "straight" to the correct solution.)

Pappus' method describes the product, not the process

So Pappus' method can neither be used to solve problems as he claims, nor can it be a reflection of how Euclid or other geometers actually worked. Since this mistake seems so obvious, then why was it made, and where did it come from? The basic mistake that Pappus made was to conflate the structures of the resulting proof, and that of the process that produced it. Most likely, this was because the method was established not from observing the actual *work* behind the proof, but only the *result* of this work—that is, the proof itself.

This can be seen by comparing Pappus' method with the structure of the geometrical proofs. The example in the figure is one of Pappus' own proofs, taken from Hintikka & Remes (1974, pp. 22f). Their account consists of Pappus' full original text with annotations; for brevity I have included only what is relevant to the present argument. Pappus' original text is in italics, and the numbered headings in small caps (2. ANALYSIS, etc.), which correspond to the elements of his method, were inserted by Hintikka & Remes. In particular, note how the steps of the analysis are repeated backward in the synthesis (cf. figure 1.5):

Hintikka's annotations:	*Pappus' original text:*
1. ENUNCIATION	
1A. THAT WHICH IS GIVEN	*Let* ABG *be a circle with the centre* E. *Let* BG *be a diameter,* *and* AD *a tangent...* *Let a straight line* JM, *parallel to* BG, *be described through* J.
1B. THE THING SOUGHT	*That* EK = EL.
2. ANALYSIS	MP = PJ. ... A, N, E, D *are on a circle.* *The angles* DAE *and* END *are both right angles.*
3. SYNTHESIS	*Because the angles* DAE *and* END *are both right angles,* A, N, E, D *are on a circle.* ... *Hence* MP = PJ. ... *hence* KE = EL.

Q.E.D.: The structure of a geometrical proof is identical to the analysis–synthesis pattern described by Pappus. Hence, it is the written presentation of such a proof that his method describes.

(Parenthetically, the many variations on design methods here find their counterpart in geometry. The schema used by Hintikka & Remes divides each of the three steps into two parts, of which I have included only 1A and 1B here; The traditional Greek schema contained six parts: *enunciation*, *setting out*, *specification*, *construction*, *proof*, and *conclusion* (*ibid.*, p. 6). Like the various design methods, Pappus' schema and these two are variations on one and the same theme.)

How come this mistake?

Put differently, the fundamental oversight in Pappus' method is that the structure of the *product* and the structure of the *process* behind it are held to be the same. It is evident that actual problem solving does not, and cannot, follow Pappus' schema.

It deserves to be said, though, that Pappus does make certain important observations. Still, there are some fundamental oversights. These have caused serious problems, in particular because they have been carried over into other domains, in those later models that draw upon Pappus original account; it would however be wrong to hold him responsible for this.

With this clear discrepancy between what Pappus writes and what really happens, one is left to wonder how such an oversight could be made: How could the structures of process and proof be conflated? Or was it ever realized that one was mistaken for the other? Especially since the difference is so very clear, and that many others have made the same mistake later. For example, Hintikka & Remes (*ibid.*) do not detect the mistake, but even write the following in the first sentence in the opening chapter:

> Analysis is *a method Greek geometers used* in looking for proofs of theorems (theoretical analysis) and for constructions to solve problems (problematical analysis). (p. 1, my italics)

This is in a book where one chapter aims to examine whether Pappus really followed the method in producing his own proofs. And as seen earlier, Polya (1945) adopted Pappus' scheme in its entirety as a central piece in his own recommendations, in a book which is expressly meant to provide concrete help and practical advice on problem solving.

2.3 The modern mistake

But the mistake, where the structures of product and process are conflated, is not an isolated instance. The same mistake has been repeated in modern times, although in a new, updated form. This shows that it cannot be written off to the primitive state of Greek science, or as a singular incident.

As stated in section 1.2, proofs do not look the same way today, due to advances in modern formal logic. The steps are no longer presented both backward and forward; now there is just one sequence going in only one direction, forward from axioms to the proven theorem (also cf. figure 1.8).

The marks of this pattern in models of rationality are found mainly in disciplines such as software engineering, engineering design, and artificial intelligence (cf. Adelson & Soloway 1985, Dasgupta 1989, Jeffries *et al.* 1981, Parnas & Clements 1986, Swartout & Balzer 1982). This is probably due to the direct and indirect links between these disciplines and formal logic. For example, a computer scientist is much more likely to have been trained in formal logic than is an architect.

The most concise evidence of the modern pattern is the quotation from Parnas & Clements given earlier:

> Ideally, we would like to *derive our programs from a statement of requirements* in the same sense that *theorems are derived from axioms in a published proof.* All of the methodologies that can be considered "top down" are the result of our desire to have a rational systematic way of designing software. (1986, my italics)

Elsewhere they also write: "Each step taken can be shown to be the best way to get to a well defined goal." This clearly refers to formal logic, where each step in a deductive chain has been formally proven to be correct. The authors also endorse a design method that follows this same pattern.

The main features of the modern version of rationality are firstly, that the distinction between analysis and synthesis has disappeared, as two phases going in opposite direction; secondly, that the process itself also starts with the axioms, and consists in deriving the theorem from them. These changes mirror the changes in the format of modern proofs.

As a result of adapting this "modernized" pattern, the design process is taken to be a progression where the design is derived from the requirements, as though the theorem were derived from the axioms, and not the other way around. Also, the requirements specification is here held equal to the axioms (cf. the Parnas & Clements quote), when it actually corresponds to the proven theorem, since the requirements specify what is "sought". This quite serious mix-up is probably due to the fact that in design methods you are to begin with the requirements, and that the modern proof schema begins with the axioms—hence, requirements and axioms are mistakenly equated with each other.

The problem with this pattern is of course that it is an even greater distortion of the genuine process. In spite of all the weaknesses of the older schema, Pappus did make the important observation that you begin by having something you want to solve or prove, and work backward from there, rather than forward; it is also significant that you usually don't know initially just what axioms or other proofs you will make use of. (The two points are also related.) Even so, from the presentation format of modern proofs, the impression you get is that this is the procedure behind them. (There are modern approaches like proof trees and natural deduction, but which have had a rather restricted impact.)

Still, even though the case here is so open and shut, it is not different *in essence* from the distortions of Pappus' original pattern. It is the same conflation of product and process—only the effects now being more exaggerated. In both cases, the form of the presentation is based on the knowledge of how to present the proof convincingly; how to make it look rational, regardless of how it was discovered.

Today, the presentation format has been even further developed, now being even more dissimilar to the process behind it. Therefore, it is now even more obvious that this is not how it was really done. (But evidently not obvious enough.) Hence, taking this as an account of the process results in a distortion of the same kind, though to a higher degree.

Normative model = descriptive expert model

The misrepresentations in these methods are of two kinds: On the one hand, they do not work as prescriptions—people don't use them because they don't work for their advertised purpose; those who actually tried them failed to reach the stated results. On the other hand, they are also inadequate as descriptions—if you study how practitioners really work, you will find what they really do to be something quite different. For all the methods in this chapter, the failures are of both kinds.

At first thought the descriptive and normative dimensions would seem not to be related, but in fact they are, for a reason which seems to apply generally: In Pappus' case, even though his method is a "how to", he states that it is the method of three authorities (Euclid, Apollonius of Perga, and Aristaeus the Elder). Hence, his method is an account of how the experts work, to the effect that you should do the same; this is also a rather forceful argument.

Therefore, the method is a description of how skilled practitioners *do* work, and its principal function is as a prescription of how others *ought to* do their work. This is also how it is possible for the method to fail in both respects: Because it is a misrepresentation of what expert practitioners really do, the detailed procedure also fails to serve as an aid for others. Hence, the normative and the descriptive are here one and the same.

2.4 The problem is in the intramental model of rationality

The underlying pattern of rational action that is shown in the Rosetta stone (figure 1.15) also substantiates the claim that the problems with the mentioned cognitive models do not originate in the computer metaphor of mind, but in this intramental model of rationality. While the standard account in cognitive science used to be that man has been made in the image of the computer, it has recently become increasingly clear (prominently by Hutchins 1995) that the converse is the true: The computer was made in the image of the human (if not the human *mind*).

It is thus the computer model of mind that has inherited the properties of the intramental schema, and so it is a mistake to attribute the problems of the classical cognitive theories to the computer model of mind; these have just *exposed* the problems they inherited from the intramental model of cognition, going back to Pappus, Euclid, and Aristotle. To avoid these problems, it is not enough to abandon information processing theory, but to completely avoid the intramental view of cognition, computerized or not.

There is also good reason to believe that this underlying model is fundamentally wrong. Firstly, it is obviously odd that it is modeled after the structure of ancient Greek geometry proofs—the bare thought is highly improbable, but the evidence is quite firm.

But an even better reason is that the brief look at how these proofs themselves are produced shows that it is seriously wrong even about them—as was shown above, being based on a quite fundamental, open *mistake*, a confusion of product and process. Had the original model of mathematical problem solving been good, then it may quite possibly have been a good start for a theory of action and cognition in general.

For the remainder of this book, the most important connection between design methodology and cognitive science is this failure, since it is located in the model of rational action; on a *cognitive* level, that is. Thereby this failure becomes of general interest; design becomes a domain where the underlying model of rationality has been put to use, under highly authentic circumstances, and failed.

The remaining chapters will follow this path, looking at design processes in order to locate and explain this cognitive failure, but more importantly, to also present a better model of design, and of sensible reasoning and action—the one that Euclid by a certain baneful craftiness kept to himself.

3. Design and cognition as inquiry

3.1 The nature of design

In this chapter I will begin to present my alternative theory of design activity. As will become evident, it draws on the analysis by Schön (e.g. 1983, 1987), which in turn owes a great deal to Dewey's theory of inquiry (1938, 1949), both of which will be discussed later in this chapter.

The view of design as problem solving has been criticized many times (e.g. Holt 1985), often on the grounds that design does not fulfill the conditions of problem solving theory (e.g. Rittel 1973). Take the earliest stages of the design process as an example. On the one hand, there is the conventional view of this as problem solving by pure analysis, based on the view of the design problem as being given. On the other hand, there are the conditions of actual designing, where design problems are anything but given, so that the designer must also do "problem setting" (Schön 1983 coined this term as a contrast to problem solving), which more or less amounts to producing also the problem itself.

The need for problem setting leads to the impossibility of separating problem definition from problem solving, which in turn provides the basis for the argument that in genuine cases, design cannot be separated into stages. The full argument constitutes the *theory of inquiry*. At the end of the chapter I contrast the resulting characterization with the stage models from chapter 1, showing how it can explain why these fail under genuine circumstances.

The design problem as given

In the conventional view, the design problem is considered as given. Most oftenly this is not stated explicitly, but is left as an unspoken assumption, as if presupposing that the problem exists before design begins (as a specification of what the problem consists in). Note how e.g. "starting from a global statement of a problem" implies that the problem is given when design begins (from Jeffries *et al.* 1981):

> The task of design involves a complex set of processes. *Starting from a global statement of a problem*, a designer must develop a pre-

> cise plan for a solution that will be realized in some concrete way (e.g., as a building or as a computer program). …
>
> Software design is the process of *translating a set of task requirements* (functional specifications) into a structured description of a computer program that will perform the task. …
>
> One can think of *the original goal-oriented specifications* as defining the properties that the solution must have. The design identifies the modules that can satisfy these properties. How these modules are to be implemented is a programming task, which follows the design task.

Newell & Simon (1972) however clearly state the problem as given:

> To have a problem implies (at least) that certain information is *given* to the problem solver: information about what is desired, under what conditions, by means of what tools and operations, starting with what initial information, and with access to what resources. (p. 73, my italics)

They do so, however, because the enumerated elements are exactly those that must be encoded into their model in advance, in order to make it work at all. As before, design methodologists are closer to reality (Parnas & Clements 1986, p. 253, my italics):

> Who writes the requirements document? *Ideally*, the requirements document would be written by the users or their representatives. *In fact, users are rarely equipped to write such a document.* Instead the software developers must produce a draft document and get it reviewed and, eventually, approved by the user representatives.
>
> … Determining the detailed requirements may well be *the most difficult part of the software design process* because there are usually no well-organized sources of information.

Hence, *ideally* problem solving theory would be correct, but in reality, producing the problem is work that the designer must do. And it is not a minor issue; it is on the contrary the *most difficult part* of the work. Note that this part is completely absent from the conventional models, since they consider the problem as given. This is obviously a large oversight that cannot be resolved by making minor adjustments to these models.

Parnas & Clements also list a number of reasons why requirements analysis cannot be isolated into an early, separate stage of the design process, the following two are among them:

> 1. In most cases the people who commission the building of a software system do not know exactly what they want and are unable to tell us all that they know.
> 2. … Many of the details only become known to us as we progress in the implementation. (p. 251)

Here they hint at some of the main points in this chapter. I will however begin to analyze the two contrasting views of constraints.

3.2 Constraints are practical

In the traditional view, constraints are parts of the problem definition, each imposing one restriction on what counts as an acceptable solution, or one *requirement* in terms of requirements specifications, which conversely can be seen as consisting of a list of constraints on the solution. It follows that constraints are regarded as given to the designer, as part of the requirements specification, before design begins. Moreover, that they make the designer's task harder by placing restrictions on her available options.

In reality however, not all constraints originate strictly in the requirements specification. Such is the case for legally imposed restrictions, such as building regulations. These still fit the traditional view in that they are laid down long before the architect begins her work, and definitely in that they make her job more difficult:

> Design legislation today may cover anything from the safety of electrical goods to the honesty of advertising or the energy consumption of buildings. … The architect today must satisfy the fire officer, the building inspector and the town planner and in addition, depending on the nature of the particular project, the housing corporation, health inspectors, Home Office inspectors, the water authority, electricity authority, the Post Office, factory inspectors and so the list goes on. (Lawson 1980, pp. 67–68)

Constraints can be both helpful and flexible

Within the design literature there are abundant examples of ways in which constraints are not the fixed restrictions given in advance that standard accounts portray them as. For example, the customary incompleteness of requirements specifications means that constraints typically *aren't* given, as mandated by design methodology. Guindon (1990b) gives a number of examples out of her protocols where the designer adds requirements that are not given in the instructions. She argues that the incomplete nature of specifications makes requirements elaboration an important task, because the constraints that the designer adds herself are often essential to a good solution:

> By simulating a Lift scenario, the designer realizes that a user may press a floor button to go in one direction, but once inside the lift, may press a lift button to go in another direction. This test case was not mentioned in the problem statement, yet it is critical for the design of a good control algorithm. (p. 288)

In the traditional view, where constraints impose restrictions, it is hard to see how *adding* constraints can be so helpful. But because specifications encountered in practice typically are incomplete, adding constraints is crucial to yielding requirements that capture the desired functionality: "The ill-structuredness of problems in the early stages of design will require structuring—inferences of new goals and evaluation criteria." (*ibid*., p. 297)

An even greater anomaly is that designers frequently impose constraints that are neither necessary nor objectively valid. They often apply them for practical (but still very good) reasons, and not from a strict necessity that is inherent in the problem. In one case, one of Guindon's subjects says:

> You would rather not have a single point of failure because if it goes down all the elevators go down. So, I'll start off thinking about a distributed control system... (p. 289)

Guindon comments,

> The designer recognizes from past knowledge with similar systems that 'no single point of failure' would be a highly desirable requirement. However, other designers might have considered low cost or high speed to be more desirable than no single point of failure. (p. 289)

In this case, the designer's professional experience suggests a distributed control solution, and as becomes evident later on, he also already knows how to implement it. Being able to apply a technique he is familiar with is probably a major reason for him to impose this particular constraint. This helps him to draw upon personal knowledge to structure his design problem, rather than as a constraint that will only make his task more complicated.

Still, having a distributed control system could just as well have been a requirement from a client. This example also shows that constraints often may be seen as belonging to the solution, as much as being part of the problem; once again, being helpful rather than problematic, quite contrary to the traditional view. Accordingly, constraints do not necessarily make the designer's job harder. Guindon sees them as being mainly very helpful (1990b, p. 290):

> ...inferred and added requirements mainly serve two purposes: (1) they lessen the incompleteness and ambiguity inherent in the specification of the requirements; and (2) they decrease the range of possible design solutions by acting as simplifying assumptions. In particular, these inferences contribute to problem structuring. Moreover they effectively guide the search of a solution by pruning a large set of possibilities.

The standard view also claims that constraints "decrease the range of possible design solutions", but that this instead *restrains* the designer instead of helping her.

The source of control principle

The standard view of constraints can be summarized thus: Constraints are restrictions on an acceptable solution that are specified in the instructions given to the designer. They are non-optional (but indeed required) and thus beyond the designer's control. It thereby seems that the notion of constraints is full of contradictions: Is a constraint helpful or a hindrance, is it fixed or optional, is it provided in advance or added during design, and is it given *to* the designer, in the problem definition, or imposed *by* the designer, entirely at her own discretion?

First of all, note that these questions make little sense within the frame of reference of problem solving theory. To such a formal model, the origin or history of its elements (such as a constraint) is of no value in reaching a solution, neither can the theory capture such aspects. And an element cannot be good or bad, only its abstract form matters. In constructing a logical proof or solving a laboratory problem, it provides no help to know who established the facts or rules, or when this was done.

The first step in resolving the apparent contradictions is to acknowledge that in real cases, unlike in theoretical analyses, all of these questions are relevant, and we therefore need to adopt a perspective that will allow us to account for them. The origin of a constraint is a point in case. In practice it does matter, because it determines how rigid or flexible the constraint is. This I would like to call the *source of control* principle: the further away from the designer the source of a constraint is located, the less control of it does the designer have, and the less flexible is the constraint (cf. Lawson 1980).

Legislated constraints are completely rigid

One end of this flexibility scale is represented by the legislator:

> … no designer would want deliberately to construct a dangerous building. However, often regulations have to be applied in situations which were not predicted when they were framed … they must be satisfied without question, and cannot be weighed against other factors and considerations. (*ibid*., pp. 67–68)

This kind of constraint imposed by laws and regulations (concerning fire safety, electricity, plumbing, etc.) are absolute and beyond the designer's influence. The legislation has usually been laid down long before the design process begins, and in an institutional context very remote from the designer's office. Therefore, the designer cannot negotiate a problematic legal constraint if it causes problems. Such a situation is not uncommon, since rule systems such as building regulations must be very generally held and cover a wide range of cases (*ibid*.). Neither are they coordinated with other regulations. It would be impossible to anticipate all possible future situations that may arise, or all the ways in which a norm may come to interact with other regulations. For this reason, the problem arises from such a constraint having been laid down in a context completely detached from the setting in which it will be applied. For the designer, there is no other option than to comply with such a norm, the source of which she has no access to. If a conflict arises, other requirements will have to be compromised instead. They are then compromised in a negotiation of conflicting constraints, not because they are less important, but because they can at all be negotiated.

Client-imposed constraints are somewhat flexible

Requirements that are imposed by clients and users represent a middle ground, being somewhat flexible. A client is guaranteed to have a number of wishes and demands on the product she is paying for. Still, if the designer finds that a requirement or a combination of them necessarily leads to a bad solution, she has the option of negotiating this requirement with the client. In particular, a user-imposed demand will have to be compromised if it conflicts with a more rigid constraint such as a legal norm.

If the client agrees to drop a certain harmful prerequisite, this will probably even lead to a more desirable outcome for herself: If she sees the conflict, she can reassess her needs and find a way out that will lead to a better result than would a blind compliance with the given requirements. After all, much of the purpose of the design process is to come to a better understanding of initially vague and unclear requirements. This understanding should be brought back to the client so that she can use it to reappraise her own needs.

Thus, in all, client-controlled constraints are negotiable and rather flexible, and this circumstance should be used to best advantage. In striving for a good end, negotiating constraints is among the best of the available means.

Designer-imposed constraints are completely flexible

On the flexible end of the scale are the constraints that the designer is completely in control of because she formulates them herself. This is the kind of constraint discussed in Guindon's example above. When her designer made a distributed control system into a major priority, this constraint was not chosen out of necessity or because the client or a law dictated it. Constraints of this kind are completely adaptable; the designer can take a totally pragmatic attitude toward them. She can select them out of her own preferences, and introduce or scrap them as she revalues their usefulness.

Hence, a constraint that is rigid and beyond the designer's control can be very problematic and can truly restrict her range of action. But it may also free her from a range of design decisions, and in that case its rigidity is not a problem. That is also the reason why constrains become so powerful under the designer's own command. A well chosen constraint can be very helpful even though not strictly necessary. By reducing too wide a range of options, it can create structure where the requirements specification is lacking. Because of their reductive function, constraints are often seen as simplifying assumptions (e.g. Guindon 1990b), and under the designer's control this is true because they can do no harm.

The source of control principle resolves the contradictions

The source of control principle can be rephrased as follows: the less the source of control over a constraint is involved in the design process, the more rigid is the constraint. This is just a clarification of Lawson's principle, where "further away from the designer" is replaced with "less involved in the design process". Although constraints are useful, there is no advantage *per se* in their being absolutely rigid and non-negotiable. A flexible constraint gives the designer the option to renegotiate it if complying with it does not lead to a good result. Such renegotiation is what it means to involve the source of control in the design process.

This is the solution to the apparent contradictions above. The source of the contradictions lay in seeing constraints as fixed and as parts of the requirements specification (or in problem solving terms, as defined by the problem statement). For example, a constraint may work as if it were given and beyond the designer's influence—but that is just the special case where the source of control precludes a constraint from being negotiated. So, these are the answers to the above questions: *Is a constraint fixed or optional?* That depends on how much the source of control is involved in the design process. *Is it provided in advance or added during design?* That also depends on the source of control. *Is it given in the problem or imposed by the designer?* It is never strictly "given", as in the traditional view. Different sources of control can make it anything from completely rigid *as if* it were given (but only as if), to designer-elected and thus entirely flexible. *Is it helpful or a hindrance?* In general a constraint is useful, by reducing complexity and adding structure. But if the effect is misdirected and the control of a bad constraint is also beyond the limits of negotiation, then it may become a hindrance.

3.3 Pragmatism & the theory of inquiry

The two perspectives on constraints that I have contrasted belong to two different modes of scientific explanation. Theories that adhere to the traditional perspective typically have nothing to say about the origins of constraints, but this in effect implies that this origin has to be unproblematic—that they have to be "given". This is the case for example when it is said that design *begins* with requirements analysis—implying that no work is required before that—then they must in effect have an independent existence, prior to design or any other activity producing the requirements, etc. And this is even more clear in problem solving theory, where everything is encoded in the problem space before the whole process starts, it is all but stated that the constraints thereby are given, among other things, to the solver.

The philosophical consequences that lie implicit in the standard view of constraints make it a so-called *realist* position. This term refers to their being "real", i.e. existing independently of someone having to "create" them; Dewey (e.g. 1949, Appendix III) used the term "ontological" instead of real, since the very idea of ontology rests on a separation between the knower and the known (the object of knowledge). I will also use his term, as it is more precise than is "real": for instance, as he argued, idealism (a vastly different position) shares the same premise, "that what is known is antecedent to the mental act of observation and inquiry" (1929, p. 23). (Note however that this use of "ontological" is non-standard, as he pointed to aspects of the concept that are not among those that are usually considered.)

Constraints as instruments

The alternative mode of explanation which the source of control principle adheres to is *pragmatism*, and it takes a fundamentally different starting point. In this view, a constraint is an *instrument*: that is, it is created for a purpose; by someone; and as a means to an end. And as an instrument it is actively *formed* to serve its purpose, by the person applying it toward this purpose.

As a consequence of this view, constraints are not fixed or static; they develop through the process which adapts them to their function. Also, they are not considered given, whatever that may exactly mean, or assumed to have an independent existence, instead they have to be created by someone, and this requires effort. And they are not objective, but neither are they arbitrary, because they have a *purpose*.

The source of control principle directly embodies the principles of pragmatism, even though Lawson (1980) who first described it made no such connection. The function of constraints as instruments is easiest to see when they are under the designer's control; then, she can freely elect and discard them, or change them. In this way, she can flexibly create a requirements specification that will serve its purpose, that is, to yield a good design solution. This is what we saw in the case of the "distributed control system" constraint above.

From the same principle, it becomes clear why rigid constraints are problematic, and why this rigidity arises when they are not under the designer's influence: when the control over a constraint is gone, the negotiability goes with it, so when it cannot be made to fit its purpose, then its role as an instrument is lost as well. The remoteness/control dimension can then reconcile all the seemingly contradictory properties, and explain the widely different kinds of constraint, by regarding them as instruments. Compared to this, the traditional realist view can account for very little about how constraints work under authentic circumstances.

Pragmatism as such

I have compared realism and pragmatism as two theoretical perspectives on constraints and the problem itself. In its essence, pragmatism is a theory of knowledge, or in the term preferred by pragmatists, *knowing*.

Pragmatism was originally the position that the meaning of a concept lies in its practical (i.e. pragmatic) consequences, and was founded by Peirce and James (e.g. Peirce 1931, James 1907). As it matured, it grew into a comprehensive, fundamental reorientation in the view of knowledge. Many of the most important developments were due to the work of Dewey (e.g. 1903, 1929, 1938, 1949). My treatment will accordingly be based on his work.

Whereas previous theories of knowledge had been based on purely philosophical issues, pragmatism eventually became a comprehensive theory based on practical matters of knowledge.

The previous "ontological" view regarded knowledge as reflecting eternal universal facts and truths, with mathematical knowledge as prototypical. To pragmatism, knowledge has the purpose of serving an individual by giving her practical and adaptive advantages; this perspective was greatly influenced by Darwin's theory. Here the empirical sciences also replaced mathematics as the model and context for scientific knowledge; the developments of relativity theory and non-Euclidean geometries became important cases in point (Dewey 1929, 1938, 1949).

Pragmatism prefers the term "knowing" to knowledge. It is a label not for a thing but a capacity, something that manifests itself in an individual's actions and which is not assumed an existence beyond that: knowing is thereby primarily an activity, and this is reflected in "knowing" being principally a verb; knowing is an entity only in a derived sense, and this is reflected in "knowing" also being a verb used as a noun, and not a noun *per se*. "Knowledge" is a noun, pointing out a thing stored in the mind.

The theory of inquiry

One major element in Dewey's contribution is his *theory of inquiry* (1938, 1949). The pragmatist view of knowledge was originally a somewhat abstract position, meant as an alternative to the equally abstract point of view in which knowledge consists of semi-linguistic, logical propositions that express ontological truths. It was a fairly general statement of knowledge being grounded in practical activity and use; although it would gradually become a more articulated position. Since pragmatism stated that knowing is grounded in use, an articulated position would also have to specify what "use" consists in. Dewey's theory of inquiry is such an articulation of the use and activity that knowing is part of.

In comparison, the previous view of knowledge was of a set of logical propositions about the physical world. That view of knowledge was also accompanied by a theory of cognition of sorts; of the processes in which that kind of knowledge is used, and put to use. This role was played by the (Fregean) theory of formal logic and deduction, i.e. the view of cognition as formal, abstract reasoning based on symbols, propositions and so on.

Since the prototypical kind of knowledge was mathematical, also the way in which such knowledge is used, i.e. in deduction, proofs and formal logic, became the paradigm for reasoning, and even the fundamental model of cognition in general. Compare this with Pappus' method being the model of mathematical reasoning and logic, as discussed in chapter 1, and it is plain to see why his method could have such thorough influence on cognitive theory.

By analogy, the theory of inquiry is the corresponding pragmatist theory of cognition. The notion of "inquiry" itself refers to those adaptive and practical, concrete activities where knowing is put to use. Cognition is held to consist in the *entire* activity of inquiry, not merely in a process of pure, abstract thinking. Also, *all* cognition consists in inquiry; it is the basic structure of cognition. (The best definitions are given in Dewey & Bentley 1949.) My use of the concept will only include the parts of Dewey's theory which are relevant for the present purposes, and it is by no means intended to be complete. However, on the points that I do include, my account goes well beyond Dewey's original version.

As witnessed in the book title *Logic: the theory of inquiry* (1938), Dewey also intended his theory as an alternative to formal logic, both as a model of scientific inquiry and of cognition in general. One must remember, though, that pragmatism and inquiry do not simply make up a set of new and different answers to the same questions that the traditional theories of knowledge are concerned with. For example, pragmatism is concerned with the use and function of knowing, which is all but irrelevant to the ontological view; accordingly, ontology is not a concern for pragmatism.

The instrumental view of knowing, as always being a means toward an end, is the innermost essence of pragmatism. Constraints, as discussed above, are one instance of this view. What it means to "treat knowing as an instrument" can be divided into two dimensions, one "logical" and one concerning process. The "logical" dimension concerns the attitude that is taken toward knowing, as having a function or purpose (versus having an independent existence and a preexisting and fixed meaning, apart from purpose and specific context of use). The process dimension concerns how knowing is "treated" in the most concrete sense: Firstly, how and when it is used for its purpose, the activities it is part of, and the specific actions that are taken there; And secondly, since it is not given, how it is created and adapted to its purpose. These processual aspects are what the theory of inquiry covers, and that I, instead of giving an account of Dewey's theory here, will introduce "as we go" in the remainder of this chapter: chiefly, these are the inquiring or cognitive function of action, the dual use/test purpose of action, and the developing dimension of inquiry.

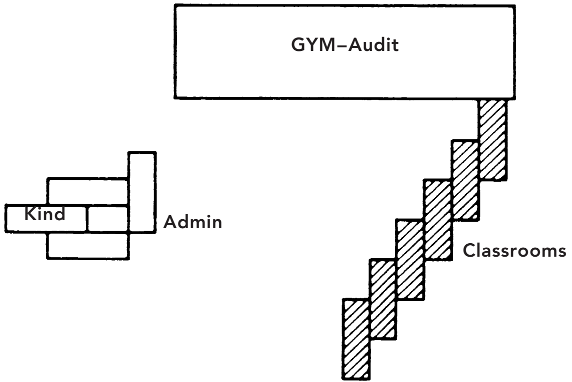


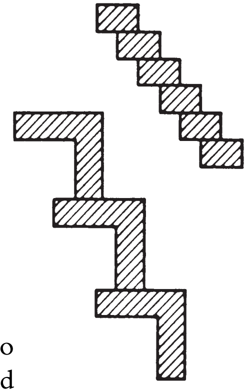
Figure 3.1 Petra's original layout.

3.4 Problem setting

I will now move on to how the problem is treated in design, and to look at a classic example of problem setting, which is Schön's own example, made famous from several of his writings (1983 ch. 3, also e.g. 1987, 1992). It describes a dialogue where an architectural student, Petra, reviews her work on a design project with her project supervisor, Quist. The review takes place at an early stage of her work.

The design review begins with Petra describing her work so far, and the problems she is having. A basic precept of architecture is that a building should be sensitive to the site it is located in, and that the physical, three-dimensional form therefore should fit into its surroundings and the location. Petra describes how she has taken this as her starting point, by trying to fit her design to a prominent land contour on the site. This is also where she has run into problems; she hasn't been able to fit the building into the slope. This issue has brought her work to a halt, and now she feels that she is seriously stuck:

> *Petra*: I am having problems getting past the diagrammatic phase —I've written down the problems on this list.
>
> I've tried to butt the shape of the building into the contours of the land there—but the shape doesn't fit into the slope. [She has a three-dimensional model which she is referring to]
>
> I chose the site because it would relate to the field there but the approach is here. So I decided the gym must be here—so I have the layout like this. [She shows a rough layout, see figure 3.1.]
>
> *Quist*: What other big problems?
>
> P: I had six of these classroom units, but they were too small in scale to do much with. So I changed them to this much more significant layout [the L shapes]. It relates one to two, three to four, and five to six grades, which is more what I wanted to do educationally anyway. ...



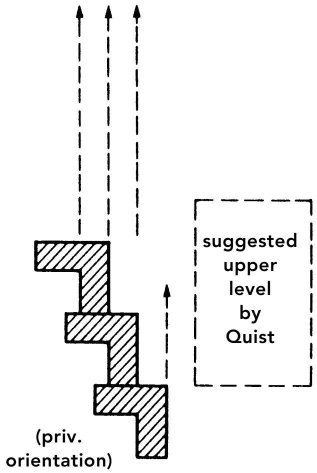
Later, as Petra's problem description becomes clearer to Quist, he enters into her description with comments, and it turns into a dialogue that revolves around the drawings in front of them:

P: This is the road coming in here, and I figured the turning circle would be somewhere here—

Q: Now this would allow you one private orientation from here and it would generate geometry in this direction. It would be a parallel...

P: Yes, I'd thought of twenty feet...

Q: You should begin with a discipline, even if it is arbitrary, because the site is so screwy—you can always break it open later.



In this last statement, Quist begins to work out his diagnosis and remedy for Petra's stuckness. Schön's analysis (based on the events that follow later in the protocol), gives the following rationale for what Quist does (1983, p. 85):

> The main problem, in Quist's view, is not of fitting the shape of the building to the slope; the site is too "screwy" for that. Instead, coherence must be given to the site in he form of a geometry—a "discipline"—that can be imposed on it.

The reason for Petra's problems is that the site in question is not suitable for adapting the building to it, which is otherwise a basic principle in architecture. This is in Quist's view why she got stuck. Instead, Quist suggests an entirely different approach. Just *because* the site is "screwy", the designer should bring with her a certain structural principle to the site, as part of her design solution. He thereby doesn't say how to solve Petra's problem—how to fit the building into the slope—instead he proposes to *change* her problem into a different one. This is what is known as "reframing", or in Schön's words, "problem setting".

The coupling of problem setting and problem solving

Here, Quist has just demonstrated an instrumental attitude toward the problem itself. When he sees how Petra is stuck, he immediately

changes the way she has set her problem. He thereby treats the framing of the problem as an instrument to adapt to the purpose at hand, here, to yield a good design solution.

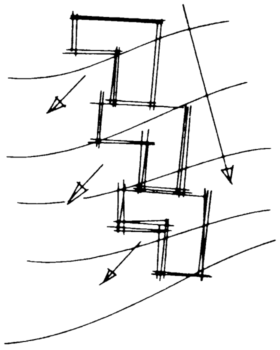
In the instrumental view, also the problem itself has a purpose. The problem statement serves to specify the function of the design solution. Hence, when the problem is not regarded as something given, it is also no longer a hindrance making life difficult for the designer. Instead, it serves to spell out the purpose of the eventual design. And the activity of problem setting becomes an inquiry into this purpose, in order to understand what it is. Thus also the task of problem setting makes a contribution to the designer's understanding.

This is seen more clearly in terms such as "requirements analysis" and "requirements specification", which correspond to "understanding the problem" in problem solving terminology. From these terms it is more obvious that this is where the function of the design is determined. Perhaps these terms are more informative because this role is more evident in design, where this activity hasn't been reduced to understanding a given problem by reading a piece of paper.

Solving is also use & implicit test of the problem-setting

When the protocol continues, Quist demonstrates the practical work that is required for successful problem setting. After having placed a transparent sheet of paper over Petra's sketches, he starts to draw over them:

Q: Now in this direction, that being the gully and that the hill, that could then be the bridge, which might generate an upper level which could drop down two ways.

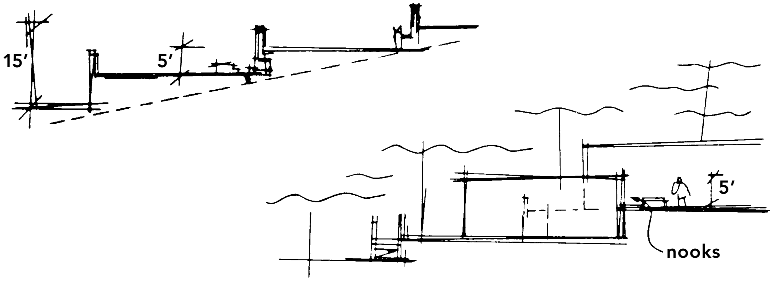


Here, Quist immediately starts to work out the implications of the new problem-setting that he has just suggested, which states that the function of the building's "external geometry" is to impose an order on the slope. He assigns this role to Petra's line of L-shaped buildings, placing them on the slope of the hill. If there is enough geometrical form inherent in this arrangement, then it will create the desired order on an otherwise "screwy" site.

Thus, when Quist starts to work out a solution, this also serves as a test of the problem-setting he has just proposed: If he can create the order that it calls for, then this shows that he has set the problem properly. At least to a certain extent, as there are many other factors that the final design will have to satisfy. On the other hand, if he like Petra fails to produce a satisfactory solution, then he should probably change the framing again.

However, Quist does not perform any explicit test; instead, it is the act of working on a solution that also serves as an implicit test of the problem being solved. Hence, also the outcome of the test is implicit. In particular, success consists in the absence of failure, showing that the problem enables him to make progress:

> Q: We get a total differential potential here from one end of the classroom to the far end of the other. There is 15 feet max, right?—so we could have as much as 5-foot intervals, which for a kid is maximum height, right? The section through here could be one of nooks in here and the differentiation between the unit and this would be at two levels.



Thus, when Quist here moves swiftly from issue to issue, one idea generating another, then this easy progress should be seen as evidence of a problem well set—no news is good news. If there were problems, then we would see them in his work not moving so smoothly, but instead becoming erratic and staggering, bumping into obstacles and not moving forward. Instead, each "right?" works as one link in a chain of possibilities.

As Schön showed, from Quist's comments we have implicit evidence that he is attending to the feedback from his actions and the emerging sketch; of whether they tell him that the solution (and thereby the problem) is good or bad. After a few more moves, he acknowledges more explicitly that his drawing has served as a test:

> Q: The kindergarten might go over here—which might indicate that the administration over here—just sort of like what you have here—then this *works slightly* with the contours—then you might carry the gallery level through and look down into here—*which is nice*. (my italics)

The comment that the solution "works slightly with the contours" refers back to his original problem-setting, stating that the solution has been brought far enough along for him to deem it as workable and "nice"—if not an unqualified success, which it is still too early to judge anyhow. He had set out to fit the layout of the building to the screwy site, and he has now found this to work somewhat, and it thus seems a viable approach.

The inseparable use/test dimension of inquiry

The rationale for Quist's actions here is given by an essential element of all inquiry, which is that action has not one but two kinds of purpose or effect, in that it works as both use and testing at the same time. When action makes *use* of some piece of knowing, it is at the same time a *test* of that knowing. In the present case, by working on a solution Quist has been applying his problem-setting to its purpose, which is to yield a good solution. The solution he has worked on, and the act of working out this solution, have therefore also served to test whether his problem-setting works.

This dual purpose of action deviates from the ordinary, common sense view where action is associated with only one function, which is to produce a certain result. I will refer to this ordinary function as the first, *productive purpose* of action. As the other purpose serves the inquiry itself, I will refer to that as the second, *inquiring* or *cognitive purpose*.

Use & test, and the dual purposes, are not separate components that mix and together become part of an inquiry. Rather, they are different types of effect of the same single action or activity, facets which become visible by taking different perspectives or points of view. Either perspective may be applied to any action, and also on any level, including activities on a larger scale.

This also applies throughout the present episode of Quist drawing, from reframing Petra's problem to concluding that it works. From the use angle, the act of drawing serves to develop a solution, with the usual purpose of moving toward a final design. But from the test point of view, it can also be seen as an extensive evaluation of Quist's problem-setting. This is what Schön (1983) refers to when he describes this passage as Quist performing a *frame experiment*. And each action that in the ordinary view serves to develop the solution fits the test interpretation equally well, serving the inquiry into the problem-setting. The two perspectives are thereby entirely complementary and symmetrical, neither being subordinate to the other.

Even though Quist's drawing episode thus serves both purposes, the productive outcome is of secondary importance in the greater context of his helping Petra; after all, Quist's job is not to produce Petra's solution, but to get her on her way again; the frame experiment serves to ascertain that his suggested remedy will be effective. The production of a solution thereby *primarily* serves the frame experiment, that is, the result may well be scrapped and still have fulfilled its purpose completely. Even the problem solving may be said to have a primarily inquiring purpose. This goes to show that the inquiring purpose of action isn't necessarily subordinate to the productive purpose, but potentially even the reverse.

Still, because of the very tight coupling between test and use, it is *necessary* to put the problem-setting to use in order to test it. For this reason, problem solving cannot be separated from problem setting, to be performed as separate activities, in separate phases of the design process. This is the single most important consequence for the present argument, as it shows why the separation made by Pappus, and carried on into contemporary theory, is simply impossible. It also shows that it is impossible even in principle, not merely for practical reasons or because of accidental circumstances.

The more general version of this argument is that all knowing must be tested by being put to use; this, in turn, so as to be adapted to its purpose. Hence developing understanding, i.e. "learning", cannot be separated from using that understanding for some purpose.

3.5 It's good to be a pragmatist

Because actions are acts of knowing, they test whether that knowing is able to serve its purpose or not, by either succeeding or failing. Hence, action has a second, inquiring purpose, and because the test is implicit, a positive outcome merely consists in the absence of failure.

Thus, there is no distinct evidence of success, nor even of a test having taken place at all. But what about a test that *fails*? In fact, nor in this case is there any explicit evidence. Instead, the action simply doesn't work: It either yields a different result or none at all; you try different solutions and new angles, but nothing works; eventually you run out of ideas and find yourself stuck. In this manner, a failed test consists in not being able to move forward, but there is no overt signal of failure.

In fact, it was such a situation that developed when Petra got stuck, and which she described to Quist. She had tried to fit the building to the site but without success, and she couldn't "get past" this problem—she had even concluded that her own problem-setting was undoable, as fitting building to slope was impossible; however, she obviously didn't realize this:

> P: I am having problems getting past the diagrammatic phase—I've written down the problems on this list.
> I've tried to butt the shape of the building into the contours of the land there—but the shape doesn't fit into the slope.

Petra acts as a realist

However, Petra's own account does not imply a failure; it is evident that she has not seen her own actions as a test, neither when this occurred, nor in her after-the-fact description to Quist.

But Petra's view is in no way in error, as there are no overt (ontological) grounds for the "action as test" interpretation, neither with respect to the outcome of the test, whether it be a success or a failure, nor even of there being a test in the first place (since the test is implicit). There is no objective circumstance that she has missed or left out.

Still, her view has consequences that cannot be denied, since her problems are due to her not seeing the test dimension: Thereby, she doesn't listen to the feedback from her own actions, which would have told her that her framing was problematic. Neither does she view the negative outcome as a *failed test*, only as a *failed solution*; and so the only remedy she sees is to *change the solution*. When she restricts her efforts to trying new solutions, she has thereby locked onto one particular problem framing, as though it were the correct one, or the only one possible.

Hence she doesn't see the problem as being testable and thereby potentially better or worse, and as her own creation that is only one among many, and that she is free to change. She instead treats the problem as though it were given, and doesn't question its validity or utility; to Petra it is *the* problem, not *her* problem.

She thereby also doesn't see that her solutions are shaped by her framing, which reflects her understanding of what a solution must achieve, and which thereby also drives her solutions in a certain direction. She doesn't see the shortcomings of her attempted solutions as opportunities to understand better what a solution should do, i.e. to frame the problem better.

The bottom line here is that there is an unmistakable pattern in Petra's behavior: she consistently demonstrates the stance of a *realist*, and this is what causes her to get stuck. By not seeing her solving also as a test, she doesn't recognize the second, inquiring purpose of action. She thereby misses the signals that indicate a problem with her framing. And by neither seeing, nor using the option to *change* her problem-setting, she treats it as if it were given, instead of regarding it as an instrument and making use of it in what she is doing. It was her inability to see that she had set her problem badly, and failing to change it, that caused her to get stuck. These are all characteristics of the realist point of view, which had some very real, negative effects on her work.

Quist acts as a pragmatist

Quist, however, consistently acts as a *pragmatist*. He acknowledges Petra's solutions as being directed by her problem-setting, and thereby sees that the deficit lies not in the failed solutions themselves, but in her problematic framing: In a sense, all solutions were doomed to fail from how she had set her problem, as the impossible task of fitting the school to a screwy slope. Here, a critical element is Quist's ability to recognize the condition of being stuck; i.e. that further attempts at a solution are wasted, and to therefore switch from solving to setting the problem.

Quist also regards the problem-setting not as given, but as being created and shaped so as to serve its purpose, and as being useful when this is done right—but also that this requires that it *be made* useful. He also seems to know that he must test his framing to find out whether it works, by trying to solve it and thereby apply it to its purpose. Thus, he thoroughly acts as a pragmatist: he treats the problem setting not as given but as an instrument that you shape to make it serve its purpose. He also recognizes the inquiring function of Petra's and his own actions, and subjects his framing to an inquiry, and so on. Above all, this makes him successful where Petra got stuck.

Good designers are pragmatists, novices are realists

Earlier I compared the realist and pragmatist views of constraints, then as two approaches to scientific explanation. What we have seen here is these attitudes being taken by the designers, not by scientist observers, and not merely as ideology, but in their concrete actions.

But also Petra's description of her problems is a realist's account of her situation, very similar to the ontological view of constraints seen earlier, neither of which is very informative: the whole pragmatist dimension is missing from her analysis, which therefore captures very little of what had happened. It merely states that she had repeatedly failed to come up with a solution to her problem. Above all, it gives no clues to her available options for action, for how to resolve the situation and move on.

Hence, the two contrasting attitudes make the whole difference between frustration and progress: Quist literally *makes* his problem solvable, whereas Petra *finds* herself stuck. The bottom line is that Quist who is the "expert" is acting as a pragmatist, whereas Petra, the "novice", acts as a realist. And as we have seen, this accounts for a great deal of his superior performance. The choice of either position is not merely a matter of ideology, but has important consequences. For pragmatists, very practical consequences.

This expert–novice difference is further supported by Lawson:

> Students of design often devote too much of their time to unimportant parts of the problem. It is easy for the inexperienced to generate almost impossible practical problems by slavishly following ill-conceived formal ideas which remain unquestioned but could quite easily be modified. One of the major roles of design tutors is to move their students around from one part of the problem to another and the job of the design student is to learn to do it for himself. (1980, p. 81)

Here, students "generating impossible problems which remain unquestioned" is precisely the failure to act pragmatically, and "move the student around" was exactly what Quist did to help Petra. Lawson also makes the same point regarding constraints: when students impose their own constraints, they do not realize that these indeed are of their own making and not something given or "found":

> It is obvious that these designer-generated constraints are comparatively flexible. If they cause too many difficulties, or just

> simply do not work out the designer is free to modify or scrap them altogether. Design students often fail to recognize this simple fact but instead continue to pit their wits endlessly and fruitlessly against insuperable problems which are largely of their own making. One of the most important skills a designer must acquire is the ability critically to evaluate his own self-imposed constraints… (pp. 70–71)

Lawson regards the pragmatist stance as "one of the most important skills" to learn in becoming a proficient designer, one who knows how to put her instruments to work. Hence, in this view design education consists in turning realist students into pragmatist designers.

Expert–novice theories and cognitive science

The expert–novice dichotomy has been very popular in cognitive science (e.g. Chi, Glaser & Farr 1988, Ericsson & Smith 1991). An early theory of expert–novice differences was that of Newell & Simon. In accordance with their theory of problem solving, they proposed that the leg up that experts have is superior general problem solving strategies, namely those embodied in the authors' theory of problem solving; means–ends analysis and so forth. However, it turned out that the very opposite was the case (Holyoak 1991): experts have domain-*specific* skills that give them their advantages. *Novices*, on the other hand, when they have nothing else to go on, fall back on these most general techniques, means–ends analysis and so forth, as their last resorts in lack of other alternatives. And these techniques turned out to be the weakest of all problem solving strategies (*ibid.*).

What we have seen here is a similar case. The performance during laboratory problem solving (with a fixed, indeed *given* problem-setting, etc.) reflects how novices work on design problems, and yields clearly inferior results for them. Expert performance under realistic conditions is quite different from what is observed during laboratory studies.

3.6 The developing dimension of inquiry

One might say, taking a step back, that Petra's failure in a broader view is that she fails to see the need for *making* her problem-setting useful. She seems to think that once she has found a problem—*the* problem it seems—then that's it. Instead, problem setting is a process where the problem is evaluated and modified if necessary, so as to adapt it to its purpose.

Petra does however recognize the need for making the *solution* useful, and she performs the necessary actions of an inquiry: her initial idea was to line up the individual classroom units in a diagonal row. She developed this idea on paper, and responded to the feedback received by forming a line of three L-units that added the missing qualities.

By doing this, and by moving on to further issues as each problem is addressed, a solution will come to develop. Eventually, when the individual actions can be seen as parts of a greater whole, then they cluster into patterns where the knowing is incrementally adapted to its purpose through the individual actions, even including the false starts and having to back up from dead ends. This is the *developing dimension* of inquiry.

So whereas Petra does perform all the necessary steps of an inquiry in order to develop the *solution*, she fails to do the same with the *problem*. She appears not to see the need for doing the *work* of producing a good problem-setting, and that this is her own responsibility.

Quist recognizes precisely this need for *making* the problem solvable: problem setting is not only the act of proposing a new framing, but the whole process whereby you test it and refine it, so as to make it useful. Petra also *proposes* a problem-setting, but she fails to subject it to inquiry as Quist does: he begins to work out a solution to test the problem-setting, but also to understand the problem better, to *make* his framing useful.

He also states that his initial framing is merely tentative: "you should begin with a discipline, even if it is arbitrary, … you can always break it open later". The initial form or quality is not critical. Since the inquiry is to ensure that the framing will eventually be useful, it can even be "arbitrary" at first, as long as it is enough to get the inquiry going.

The point of view of this as an inquiry thereby shifts the emphasis from the initial proposal, whose importance is played down considerably, to the *eventual outcome*; the product of inquiry.

The interplay between problem and solution

But because of the use–test duality, there is a reciprocal relation between developing the problem and solution, so that working on either also serves to develop your understanding of the other, and vice versa. Therefore problem and solution are intimately connected and develop in parallel. This is why actual design work does not separate these two aspects from each other: on the one hand, it has proven impossible to separate them; but on the other hand, dealing with them together yields important advantages.

This can be illustrated by a study done by Nardi and others (1991, 1993), of an activity located in the border zone between design and small-scale problem solving, the activity of developing computer spreadsheet models. The first example is of an accountant who learned to develop such models himself, instead of having the company's programmers do it for him, as had been originally intended. The main reason why he did so lies in the close coupling between problem setting and problem solving. He found it impossible to describe to a programmer just what it was that he wanted. Instead, when he built the models himself, he could use the spreadsheets to develop his own understanding of what he wanted by working on the problem. That is, he wasn't able to formulate a problem statement without working on a solution to it—while doing so in contrast helped him considerably (1993, p. 13):

> *Jeremy*: We had to have rather large complex spreadsheets [for the business plans] where you had lots of variables. *And I found it easier to develop that myself than to go to somebody and say here's what I want, here's what I want, here's what I want.* And that's what really got me going on [spreadsheets] ...
>
> *Interviewer*: Why was it easier for you to do this yourself than to specify it for a programmer?
>
> J: I think it was easier because I felt that I was learning as I went, *as I was developing the spreadsheets, I was learning about all the variables that I needed to think about*. It was [as] much a prop for myself as [a way of] ... getting the outcome ... And there were a lot of false endings, I should say, not false starts. I'd get to the end and think, "I'm done," and I'd look at it and I'd say, "No, I'm not, because I've forgotten about one thing or the other."

First of all, the accountant states that he did not have a clear picture of what the problem was, even to himself, and much less one that he could give to the programmer. It seems paradoxical, but in the beginning he appeared to have a clearer picture of the *solution* he wanted than of the problem. This goes completely counter to the conventional view, but it makes sense in the way that he describes it. He couldn't make use of a programmer because it would require that problem setting *could* be separated from solving; not only between different phases, but between different *people*, even. It would have required that stage models worked as intended.

Instead he needed to develop a solution to understand what he wanted; spreadsheets enabled him to do this, and that was why he liked them. And when he describes his work on the spreadsheets as a "prop" for himself, he is referring directly to the second, *inquiring* function of this work, in that it also serves as a prop for his own thinking (about the problem), i.e. the inquiry, "as much as a way of getting the outcome". This last phrase is also a clear reference to the first, ordinary purpose of his work, that of producing the spreadsheet model itself. The way in which these spreadsheets are used parallels Quist's use of sketching to articulate his problem-setting, on point by point.

Lastly, he describes the dialectical structure of this work, which seamlessly shifts back and forth between problem and solution. It is clear that understanding the problem helps in solving it, but here the reverse is also obvious: a solution is not the end of the process, but only a false ending, as new solutions repeatedly serve to make him discover new aspects of the problem. Thus both develop in parallel, each as a "prop" for the other.

> In the study we found that spreadsheet users are very aware of the fact that their initial problem formulations are likely to be fuzzy, incomplete and badly structured. They like spreadsheet software because it helps them to work through these difficulties. (*ibid.*, p. 11)

Similarly, in a study of architects, as reported by Lawson (1980), Eastman (1970) showed "how the designers explored the problem through a series of attempts to create solutions", and found "no meaningful division" between analysis and synthesis, but rather "a simultaneous learning about the nature of the problem and the range of possible solutions". The designers "discovered much more about the problem as they critically evaluated their own solutions" (*ibid.*). This is a well-known phenomenon in design work.

In the second example, another accountant describes how she developed spreadsheets for an executive in her firm, and how the spreadsheets worked as props for her boss. Her story seems to be taken right out of a "Dilbert" cartoon:

> Oh, this [spreadsheet] is what I gave to the CFO at first just comparing Q2 [Quarter 2] year-to-date budget to Q2 year-to-date actuals. And he said, "Well, for the board meeting I want [some other things]." Every time you do this he wants it differently. So I can't anticipate it. I just give him what I think [he wants] and then he says, "Ah, no, well, I want to have projected Q3 and projected Q4, and then total projected, and then the whole year's plan on there." (Nardi & Zarmer 1993, p. 14)

By merely seeing the model, the chief financial officer can better describe what he wants. Again, the early versions mainly serve to inquire into what he wants; hard work on these is probably wasted ("Every time you do this he wants it differently"). They should merely set in motion the process which will ensure the final quality. Progress is incremental, with new problems and solutions alternating as the successive frames in a comic strip, the untiring accountant walking to the chief's office and back again. It is also evident that the task of problem setting spans the whole process, the CFO not knowing what he wants till it is on his desk. This example also demonstrates what the accountant in the previous example gained, by doing both parts of the work himself.

3.7 No pure analysis

We can now return to the question of why design methods don't work. The answer supplied by the theory of inquiry is that there can be *no pure analysis*: the duty that has been assigned to the analysis phase cannot be performed by analysis alone; it needs to be performed together with the other activities of the design process: understanding the problem, working on solutions, and evaluating your work. Compare this with the following answer to the question of what was learned from the failure of design methods (Rittel 1972):

> ...that the design process is not considered to be a sequence of activities that are pretty well defined and that are carried through one after the other like "understand the problem, collect information, analyze information, synthesize, decide", and so on; and another being the insight that you cannot understand the problem without having a concept of the solution in mind; and that you cannot gather information meaningfully unless you have understood the problem but that you cannot understand the problem without information about it...

Here, the sequence of activities is stated as the basic failure, but as I have shown in chapter 1, it is not primarily the imposed order among the activities that is the culprit, as has been previously thought, but rather the separation between them, and from which also the ordering follows. This also includes Rittel's second point, the impossibility of keeping problem and solution apart, either logically or as separate processes.

As Swartout and Balzer state, this separation is a fundamental element of the rational models of action (as seen in the Rosetta stone, figure 1.15, specification and implementation correspond to analysis and synthesis):

> For several years we and others have been carefully pointing out how important it is to separate specification from implementation. In this view, one first completely specifies a system in a formal language at a high level of abstraction in an implementation-free manner. Then, as a separate phase, the implementation issues are considered and a program realizing the specification is produced. ... *all current software methodologies have adopted a common model that separates specification from implementation*. ...
>
> Unfortunately, this model is overly naive, and does not match reality. Specification and implementation are, in fact, *intimately intertwined*... (1982, p. 438, my italics)

Again, the clash between the ideal view of how things ought to be and how they actually are. Also Guindon has documented a study of software engineers in a series of papers (1987, 1988, 1989, 1990 a,b, 1992), showing in detail that the design process does not follow the pattern of "structured design methods" such as the waterfall model, but that it instead follows an "opportunistic" pattern that deviates from the norm. But she also showed that this was not a "failure" to use these methods, but instead that there is good reason not to do so (esp. 1990a, also cf. Hayes-Roth & Hayes-Roth 1979).

4 phases vs. 4 aspects of inquiry

The full schema of the rational models of action contain three major phases besides analysis: understanding the problem, synthesis/action, and evaluation. The theory of inquiry also contains three basic aspects: the two elements of the use–test pair, and the dimension of developing knowing. Note the close parallels between the elements of each model: between use and action, test and evaluation, and between developing knowing and understanding the problem.

The two in the last pair both address the need for "learning", roughly, what you need to know to solve the problem—although they take entirely different approaches to how this is done. The other two pairs are also parallel in their function, along with analysis/thinking. (In case there is any doubt, e.g. problem solving theory explicitly states that problem solving is performed by this part alone, see Newell & Simon 1972, Simon 1981.) While design methods place them after each other and in separate phases, according to the theory of inquiry they are inseparable. They are not even distinct parts but only different points of view that can be taken; potentially even of the same, single action.

Hence, the claim that there can be no pure analysis means that if the analysis is separated from *either* of the three other functions, it cannot do the work assigned to it; the theory of inquiry also explains why this is impossible, for each of the three "auxiliary" functions.

The main point with the discussion of problem setting was that the problem cannot be determined or fixated before the process of solving it begins, but instead that understanding and defining the problem amounts to a major part of the whole problem solving task.

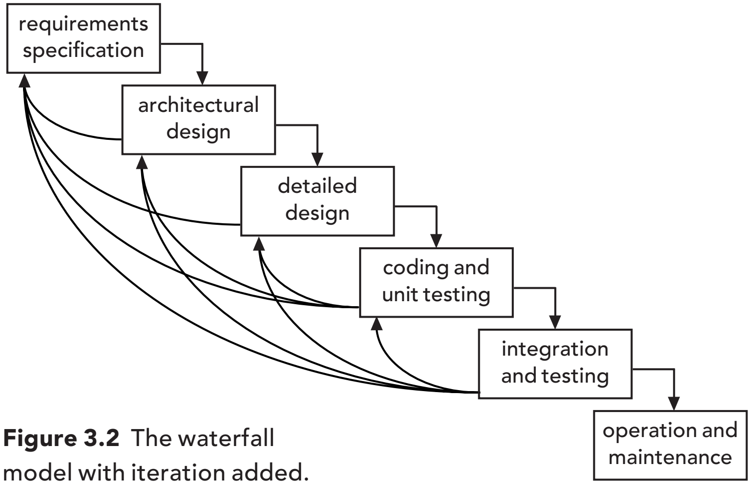


Figure 3.2 The waterfall model with iteration added.

And Nardi's examples, among others, showed that separating the two is undoable. The result of this is that setting and solving proceed in parallel, intimately intertwined and throughout the whole process, so that problem statement and solution are completed at the same time.

The simultaneous use and testing of your developing knowing, which is essential to working on problem and solution, complete the picture by showing how also action/synthesis and evaluation are inseparable from the other two components. Hence all the four elements of inquiry are tied together in a very fundamental manner. This is also corroborated by Parnas, who stated that both specification and modular decomposition (i.e. analysis) rely on an implementation process having already been done (Parnas 1985, quoted in Budde *et al.* 1992, my italics):

> There is, however, no method available for ensuring that a specification is complete and correct. *Only if a similar system has already been built* can a further system be consistently specified in advance with any amount of certainty.
>
> ...The decomposition of the overall system into modules is a simple matter in cases where the design decisions arising during implementation of these modules are known. This can only be the case, though, *if a similar implementation process has already been successfully carried out*.

That is, the prescribed procedures only work if you already have the answer, because you have already done the work once before.

Iteration

Someone might in defense of phase models refer to *iteration* as a well documented phenomenon (e.g. Carroll, Thomas & Malhotra 1979, Malhotra *et al.* 1980, Thomas & Carroll 1979). Adding iteration to a model means that you allow for the included phases to be repeated; this is represented by adding backward arrows to a box diagram (cf. figure 3.2, and also figure 1.16). However, on closer consideration, this supports the position I advocate. The reason is that iteration is a prototypical *ad hoc* extension, that is, an ill-considered added feature that handles a certain condition, but which in doing so goes against the original idea, and is therefore incompatible with it—thereby, in reality it constitutes no solution at all.

By allowing for iteration, a stage model comes to saying that you can do anything, in any order, as many or as few times as you like. By allowing for everything, it no longer says anything about their order. But if you do that, you have given up what was the purpose of these models in the first place: to specify what things to do, when to do them, and in what order, so as to guide the designer. The only substance that remains is a list of the activities that are included.

And if a design method is such a list only, then you arrive at what I have stated—that design consists of several component functions that cannot be held apart, and that display no general ordering principle among them. Hence, the idea of iteration goes to say that the original idea of separated and ordered activities does not hold. In severe cases, an *ad hoc* extension can even divest the basic model of its original purpose, and this is what iteration does.

The cognitive roles of the 3 "auxiliary" activities

As a result, the roles assigned to the three "auxiliary" activities—evaluation, action/synthesis, and understanding—differ sharply between the stage models and inquiry theory. In the stage models, these are all downgraded to mechanical, empty, and meaningless activities, with the result that all the important work is pushed back inside the analysis. The most obvious example is the final evaluation stage, which can make no contribution at all to the design, as it is performed only when the design has already been completed. At most, evaluation is held valuable for *future* designs, which may draw on the experiences gained from the evaluation. Just such a role is also given to evaluation in problem solving by Polya (1945).

In inquiry theory, in contrast, evaluation has a quite crucial role, leveraging each successive attempt by enabling it to draw on the experience from previous trials; experience which evaluation generates by drawing out consequences and lessons learned from these attempts. It *can* have this role, as it is performed concurrently with the other functions. The role is directly reflected in the notion of *formative evaluation*, i.e. of the kind that serves the formative stages of design; such practice has increasingly been lifted forward as an invaluable but undervalued design technique (e.g. Carroll 1997, Hix & Hartson 1993).

Similarly, action/synthesis is an entirely mechanical phase which can have no secondary, inquiring function, since it via the plan is completely predetermined by the analysis/planning phase. And since it is done only after the analysis has been completed, it cannot have any inquiring function.

And finally, understanding the problem is reduced to reading a given problem statement; compare with the previous statements of this as the hardest, most important and most difficult work of real design. This is also why phase models require that all information be *given*; pure perception or "input" is not capable of *generating* all this required information by itself; that takes *inquiry*—however, if it were given, reading alone might suffice. Compare with the Parnas quote (p. 98) stating that a complete specification is impossible to produce with less than having previously made a full implementation.

The need for intramental magic

Having trivialized these three other functions, the separated models push back all the important work into the analysis part; the intramental "black box" thereby seems to require magical powers when the whole task of design is assigned to it alone; hence, the air of mystery surrounding the concept of "creativity". In the inquiry model, the other three functions can make important contributions, and the burden on the mental part, and the need for magic, diminish accordingly. It seems far less puzzling that an architect can come up with an elegant, creative solution by diligently working at his drawing board, if she were held to do this just by having a creative idea, or sitting down to "think out" the same thing.

There is a saying that genius is 1% inspiration and 99% transpiration. If this is compared to the work of inquiry, the ninety-nine percent of non-intramental transpiration seem much less a waste of time on false leads, than an essential part of the work involved; the remaining percent then appears less wizardly. In fact, Thomas Alva Edison, who is accredited with this saying, apparently made hundreds of discarded "solution attempts" before building the one that would become the light bulb. We all know that we use less than ten percent of our brains' capacity; still, to claim that the non-mental parts stand for 99% of our cognitive capacity would be to go just *slightly* too far.

Cognition is inquiry, not intramental thinking

In summary, the roles of the three other components of inquiry are the main reason why the separated models are so weak; they fail because the unit doing the cognition has been deprived of the important services *to cognition* that these other functions provide. Without these contributions, an isolated, intramental, "pure" analysis is made powerless.

Hence, not merely analysis produces the proof/plan/solution by itself, as the rational action models state, but all the four components of inquiry together are required for doing this; whether it concerns geometrical proofs, design, or other problem solving. And this is my general point about cognition, too: it does not consist of "pure thinking", but of *inquiry*, including all the four aspects I have enumerated here.

And now, having so far devoted an unproportionate amount of attention to the mental aspect, the rest of the book will be devoted to the cognitive roles of the other, shall we say, 98%?

4. The cognitive roles of action and world

> Thinking is one of the most notoriously intractable parts of psychology since the thought process is not easily observed. ... The designer, however, has never resembled Rodin's "Thinker" who sits in solitary meditation, but has in contrast always externalized his thoughts, not only as an end-product in the form of a design, but as an integral part of the process itself in the form of drawings and sketches. (Lawson 1980, p. 96)

4.1 Introduction

The topic of this book can be formulated as a question: Why do designers work the way they do, when the traditional theories of cognition and design say that designers should be doing something quite different? This chapter and the next bring the question down to the level of actual action, looking at what happens moment by moment, when the designer is sitting there working on her design, with pencil, clay, balsa wood, or whatever, in hand. On this level, the question becomes: Why do designers work out their designs physically, in the world, when the cognitive theories we have say that design should be done in the head? The starting point here is that conventional wisdom in cognitive science holds mental simulation, planning, etc. to be vastly superior to physically working on a problem, because it allows you to make predictions, test alternatives, and so forth. So why do designers not do what cognitive scientists say they should?

The answer I present in this chapter says that cognition is not an activity going on inside the mind, but an *interactive* process between mind and world. I present a theory that I tentatively call *interactive cognition* which is an effort to explain why designers do what they do, on this "action" level. My main point is to show that an interactive cognition has important advantages over following intramental principles, being both simpler and more effective at the same time.

Sketching

A prototypical design activity to explain in this way is *sketching*: It takes place in a simple setting and with only very simple tools; with the designer sitting at her desk with paper and pencil. As it turns out, there is little need for more sophisticated technology, because there is little emphasis on the resulting sketch as such—sketching should not be mistaken for the process that produces the final drawings, the detailed and carefully produced drawings that will be the final result of the designer's work. The aesthetic impression of sketches is therefore often far from the sophistication and elegance of the drawings we usually associate with design; in this respect the one on the title page (p. ii) is unrepresentatively beautiful (also see figures 6.1–6.3, & Black 1990). Robbins 1994 contains drawings of the latter kind.

Instead, sketching is the process by which the designer works on her problem, and as such it serves several purposes. She sketches to understand her design problem and what it requires of her, to explore its particular circumstances and problems that must be tackled, to experiment with different approaches to a solution, and to eventually work out her final design, among other purposes. Therefore, the emphasis is on the process of sketching in itself rather than on the product, and for the process paper and pencil are highly suitable, and in their own respect quite sophisticated tools.

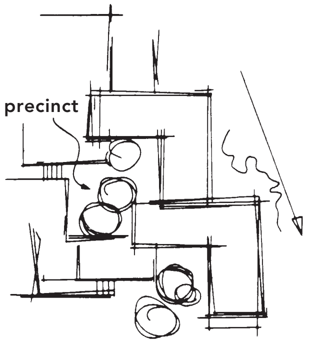
Beside sketching, there is a whole ecology of design techniques; this topic is covered in chapter 6. Still, sketching with pen and paper is widely regarded as a characteristic design activity, often even as the very essence of what design is about (e.g. Schön 1983, 1987). This is partly because it is so ubiquitous and typical of design in its various forms; architecture, industrial and graphic design, and so on, and partly since it is representative also of the other techniques: Architects use models of building sites and buildings, as for example Petra had done in Schön's protocol which I will discuss again below. Prototypes of the developing product are ubiquitous among industrial designers. The various techniques exist because their differences make them variously suitable for different types of task: While paper and pen are useful for designing floor plans, other media express tactile and three-dimensional qualities better, for example.

Quist's demonstration of sketching

To illustrate what designers' sketching is like, I will return to the episode with Quist and Petra from chapter 3 (Schön 1983), where I analyzed the inquiring structure of Quist's actions. Petra had set her problem as trying to fit the school building into the slope of the hill, but hadn't been able to solve it. Quist instead reframed the problem as imposing a geometry of his own, to bring "discipline" to the site:

Q: You should begin with a discipline, even if it is arbitrary, because the site is so screwy—you can always break it open later.

After having done this, Quist starts to develop a solution to the problem he has set. He does so by starting to sketch over Petra's drawings to work out the consequences of his framing, at the same time describing to Petra what he is doing:

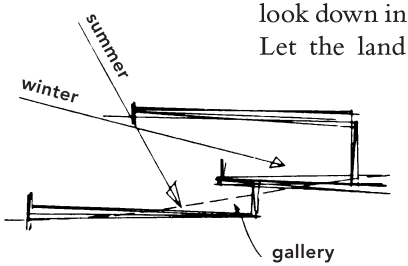


Q: Now in this direction, that being the gully and that the hill, that could then be the bridge, which might generate an upper level which could drop down two ways.

We get a total differential potential here from one end of classroom to far end of the other. There is 15 feet max, right?—so we could have as much as 5-foot intervals, which for a kid is maximum height, right? The section through here could be one of nooks in here and the differentiation between the unit and this would be at two levels.

Now you would give preference to that as a precinct which opens out into here and into here and then, of course, we'd have a wall—on the inside there could be a wall or steps to relate in downward. Well, that either happens here or here, and you'll have to investigate which way it should or can go. If it happens this way, the gallery is northwards—but I think the gallery might be a kind of garden—a sort of soft back area to these.

The kindergarten might go over here—which might indicate that the administration over here—just sort of like what you have here—then this works slightly with the contours—then you might carry the gallery level through and look down into here—which is nice.



Let the land generate some sub-ideas here, which could be very nice. Maybe the cafeteria needn't be such a formal function—maybe it could come into here to get summer sun here and winter here.

(Also cf. the figures in chapter 3; unfortunately only redrawn versions of the figures are provided with Schön's texts.) When Quist here arranges the L-units on a line down along the slope, he determines the locations of these units in a manner that is typical of sketching. According to an intramental account, he would conceive their placement first, and then afterward make a sketch to document his decision. But the protocol does not bear this out. Instead of following such a distinct and simple pattern—first thinking and then drawing—Quist's actions make up a more intricate structure.

Schön (*ibid.*) called this pattern *moving–seeing–moving.* From Quist's descriptions of what he draws, we can see that his solution develops as he is drawing. When he starts to draw, he makes a first "move", imposing the geometry of the three "L":s on the slope. In this case, the move originates in his framing. This move then allows him to "see" or visually appreciate the consequences of his move on paper, and hence of his reframing. Then this appreciation informs the next move, and so on. In this way he continues to work out his ideas by sketching, step by step, where each step suggests what the next step might be. The cycles of seeing and moving repeat, they are incremental by nature—hence the term moving–seeing–moving.

The way in which Quist verbally connects his descriptions of what he is doing also shows that his own process of drawing makes him think and have new ideas: Expressions like *that could then be…*, *which might generate…*, and *which might indicate that…* attest to his stepwise reasoning-by-drawing. He also says *There is 15 feet max, right?—so we could…*, and so on. These connecting phrases indicate that he is sketching to work out a solution, not merely to record his progress.

What he draws is then clearly not just the "output" of something he has already conceived in his mind; his words are not an after-the-fact report of something he has already thought out. The increments instead indicate that his reasoning takes place as he is drawing. He is using the drawing and the seeing as the basis for his next move, using visual feedback instead of trying to visualize each step in his mind's eye, and he uses physical drawing of concrete solutions instead of abstract reasoning about requirements and constraints.

The protocol thereby gives no grounds for making a clear distinction between thinking and drawing. The intricate pattern of Quist's activity gives little justification for treating them as separate activities, but rather as two aspects of one single activity. Thinking and sketching go on in parallel and mutually enable one another to move forward. Drawing enables thinking to proceed and vice versa. It is the entire physical activity of sketching, not merely thinking but the whole *inquiry*, which serves the function that we usually call reasoning. This is the means by which Quist develops a solution, and a *function* that we generally regard as cognitive. This function does not become less cognitive by involving other elements than pure thought.

When he comments that his solution "works slightly with the contours", he is referring back to his original problem-setting. In Schön's words, he has been conducting a drawing experiment, and this comment shows that he is at least partly satisfied with the outcome of his experiment.

Quist also instructs Petra to use drawing experiments in the same manner. He tells her to "investigate which way it should or can go". He says that she "might carry the gallery level through", implying that she will have to try it to find out.

The aspects to make sense of

In summary, there are a number of points about sketching that we need to make sense of. One main question is why designers work in this way; my answer is that it provides definitive advantages over the intramental style. This translates into some more specific questions:

- What does sketching do that you can't do intramentally?
- What are the advantages; how can there be advantages?
- How do activities and materials contribute to cognition?
- How can they contribute; how can they have a cognitive function? What does it mean that they contribute?

A second problem concerns the structure of sketching activity, and why it looks like it does:

- How can sketching have the highly integrated role that it seems to have in the *cognitive* process of developing a solution?
- How can you explain that there seems to be one single process, sketching, going on; not two, sketching and thinking?
- What are the functions of the stepwise dialectical and moving–seeing–moving structures?
- What are the functions of drawing experiments and "investigations"?

As it happens, these features of the working process contain the seeds for answering the above questions.

4.2 Conversation vs. writing

Before going into the theory in full detail, I want to present a preliminary idea of what interactive cognition is, and how it can have important advantages over intramental cognition. I will do so by making an analogy with two ways in which people use language: spoken conversation, and written communication via an intermediating text.

In this analogy, interactive cognition corresponds to ordinary, everyday conversation, where both speaker and recipient are co-present when the communication takes place—this corresponds to an interactive cognition (speaker) having access to the world ("recipient"). The intramental model corresponds to writing. There, the writer (intramental cognition) typically produces her message without having the eventual reader (world) co-present with her at the time of writing. (And symmetrically, the person writing is not available to the reader when she eventually gets the message.) This reflects that in the intramental model, cognition operates as if it were completely isolated and remote from the world and thus its object of concern, even in those cases when it is readily available and close at hand, physically speaking.

There are two reasons why I use this analogy here, and will keep returning to it in what follows: The first is that ordinary language use is a domain that everyone can relate to, regardless of whether they are familiar with linguistics or not. The second is that conversation is practically the only domain where interaction has been studied, so that there is some scientific work that I can draw upon. However, as this is merely an analogy, I have taken care to avoid the aspects of conversation that have no correspondence in interactive cognition, e.g. the other party having a cognitive capacity independent of the speaker, and her ability to take an active role in the interaction.

As it happens, the traditional model of speech production is based on the model rational action from chapter 1, which is intramental in itself. Thus, speech in this view consists in output, controlled by a plan, which is derived from an intention. Hence, no feedback or influence from the recipient goes into speech production. This shows the connection between writing and intramental cognition, where a model already exists; it then falls on myself to provide a corresponding model of interaction/conversation.

In an argument similar to the present one, Clark & Wilkes-Gibbs (1986) pointed out that the writing-based model doesn't apply to ordinary spoken language, since it works as if the recipient weren't even there. Thus, it assumes that also co-present conversation works as if it were writing, where no reader is available. The authors therefore called this the "literary model" of language production.

The same thing can be said about intramental theories of cognition: They make the same assumptions about cognition: it works as though the world weren't there, even when it is:

> For the crucial activities, at least of human problem solving of any complexity, ... take place centrally. *This is true even when the desired object and the required activity are physical.*
>
> (Newell & Simon 1972, p. 72, my italics)

Here I will be making an argument similar to that of Clark & Wilkes-Gibbs: Cognition does not work according to the "literary" model if it doesn't have to, just as conversation does not.

For the sake of comparison, consider what a person has to do to communicate: Her problem is to make sure that her message will make sense to the addressee. The problem for a *writer* is that her eventual addressee will be at a remote time and place when she reads the message. Because the writer is *separated* from her addressee in this manner, she can neither find out, nor ensure whether the text will indeed be understood when it is read.

Often, the writer does not even know *who* the eventual reader will be. Therefore she must try to make a prediction of who her audience will be, and adapt her message to that prediction instead. Moreover, even if she does know who her eventual reader is, she still cannot predict very accurately what will make sense to her and what will not. So the writer can neither find out nor know for sure that her text will have its desired effect.

For a *speaker* the situation is entirely different, because the listener is present with her right then and there, as she is producing her message. Thereby she can communicate the same message in a completely different manner. First of all, the speaker can get feedback directly from her addressee. Thereby she *can* find out whether what she says makes sense or not. Thus, she can also know for sure that her message works as it should, and so she doesn't need to make any predictions as the writer does.

On the one hand, the conditions for written communication here are analogous to the basic structure of intramental cognition: Instead of operating directly on the world, it uses a representation as a stand-in. The representation is a substitute like the writer's estimate of her reader, and for the same reason it adds a layer of uncertainty. On the other hand, just like a speaker, an interactive cognition instead uses feedback from the source by interacting directly with the world. Thereby it attains higher certainty with less work.

Intramental cognition can of course use such direct feedback, too; that is, without having to make predictions and estimates. But to do so requires the same capabilities as interactive cognition uses, in addition to the representational system—which then just adds extra effort without any gain. It becomes a superfluous intermediate step, standing between individual and world. Hence it is a literary model of cognition: It treats the environment as if it were not directly available even when it is, just as the literary model of speech works as if the listener were not there even when she is.

I would like to stress that the relevant distinction is not between written and spoken language *per se*, but whether or not the speaker/writer has the recipient directly available to her. There are situations where speakers are remote from their addressees, and others where writers do have their audience with them. Still, the prototypical circumstances for speaking and writing capture the relevant distinction in an intuitive fashion.

In addition to making predictions, a writer must deal with her uncertainty by making *compensations* in the message. Typically, this compensation amounts to making background information and context explicit in the written text. This will make her text longer, typically a great deal longer than it strictly would have to be. This may be understandable if we consider that the writer cannot after the fact make up for misunderstandings or problems that occur during reading. Such problems will have to be averted in advance, in the text itself, but still without guarantees of success.

In the literary model of cognition, these compensations correspond to the elaborate plans and action sequences that an intramental cognition must prepare, since it has to provide for all eventualities that may occur—and then some more, to be on the safe side.

Because a speaker has her listener available right in front of her, she doesn't stop at passively listening to whatever feedback the listener will give her; there is also a second purpose with her speech. Beside the obvious function of saying what she wants to say, she can also use her speech to *actively inquire into* her listener's understanding. She can actively make the listener give her the feedback she wants. This way she can determine much more precisely and directly whether her message works the way she wants it to, getting precisely the information she needs, instead of being content with whatever response the listener gives her. Asking questions is the most obvious way of doing this, but as we will see there are other ways that are much more sophisticated.

In written communication, the different activities are clearly grouped into separate phases, happening at different points in time: first writing, then reading. Conversation however does not consist of such phases, where first the speaker's questions explore the addressee's background knowledge, then followed first by the listener's answers, and then the speaker giving the necessary background information that the addressee lacks, before finally delivering the message itself. Conversation is not divided into such large separate chunks, with different roles for the participants in each of them, for instance with regard to who speaks and who listens. The power of everyday, informal conversation lies precisely in that it *isn't* restricted by the separation that makes writing problematic. The ability to alternate back and forth is what makes the compensatory measures of writing literally redundant in conversation. And the more often the speaker gets feedback, the less must she be concerned with getting her speech right the first time, since she will know right away if there is a problem. And if the listener can interrupt at any time, then less time will be wasted on talk that doesn't make sense to her.

Therefore, in conversation the participants' roles are much more democratically assigned than in written communication, with frequent chances for each party to take the floor throughout the conversation, and with shifts between speakers occurring frequently (Sacks, Schegloff & Jefferson 1974). In conversation, parties tend to speak in small, brief contributions, and the floor alternates between speakers frequently in an intricate yet smoothly coordinated web of contributions from several participants (*ibid.*).

Finally, there is no need for a speaker to anticipate and try to prevent any trouble that an addressee might have, since she can give immediate feedback if it happens, and the speaker can then make the necessary adjustments accordingly. Instead of starting by providing a great amount of background, a speaker can therefore be radically concise and to the point, backing up and becoming elaborate only if and when she has to, and even then no more than necessary. For example, in the following passage from a casual dinner conversation, a very terse question only requires a small elaboration to be understood properly (Tannen 1984):

A: Do you *read?*
(1.0)
B: Do I *read?*
(0.5)
A: Do you read *books?*

The result of the interactive relation between participants in conversation is that communication becomes much more efficient. There is no need for supplanting each phrase with an introductory lecture, since it is straightforward and simple to determine what needs to be said. A speaker can find out whether she is making herself understood, and also make sure and be certain that she does, neither of which a writer can. And all of this with fewer words than a writer would need. For these reasons, conversation can achieve a higher quality than can writing, all the while doing less work, using fewer words, and in shorter time.

The comparison I have presented here implies that having cognition connect directly with the world, without intermediary representations of it, leads to greater certainty. In addition to that, it also suggests that action can have a second purpose beside the usual one of producing the desired result, just like talk in conversation: Manipulating the environment will give cognition richer and more relevant feedback. This makes a successful outcome not only more likely, but also easier to reach.

Further, the separation of intramental cognition from input/perception and output/action would in interactive cognition be replaced by fine-grained, intertwined interaction. In this way, the activity can evolve in close coordination with the world, obviating redundancy and making it lean and adapted to the specific circumstances. Separated, "literary" cognition, not being able to get any feedback, must instead produce the elaborate, redundant, better-safe-than-sorry action schemes that are necessary to provide for all eventualities. 'Interactive' actions will not have to be carefully crafted from the beginning, since they can swiftly adapt to feedback.

So in summary, the efficiency of conversation translates into promising prospects for a similarly organized, interactive cognition. It would enable an individual to act with more precision and greater success, while at the same time using considerably less effort, and doing so in fewer words. Without having seen that this is possible, and does happen in the case of conversation vs. writing, it would seem impossible for interactive cognition to improve quality and decrease effort at the same time compared to the traditional, intramental, mode of operation.

The advantages of interaction in the properties of conversation translate into the corresponding cognitive organization that I will call interactive cognition. In order to work out the details of this scheme, we first need to look at the concept of interaction as such.

4.3 Preliminaries: The meaning of 'interactive'

According to a dictionary, interaction means "mutual or reciprocal action or influence". *Interactive cognition* is meant to indicate that mind, action, and world *mutually* determine an individual's doings, in interaction. Cognition, of course, effects changes upon the world. In conventional cognitive theory, it is the mind alone that determines what an individual does, in a simple causal relation. In interactive cognition, the cognizing individual on the one hand, and the world on the other, *reciprocally* influence each other. In other words, mind and world interactively determine each other, and in particular they interactively determine cognitive performance.

The essence of interaction is that both (or all) participants give and take; speak and listen; act and perceive. This mutual influence is what breaks up simple causal schemes. A second point is that the traffic back and forth comes in a frequent exchange of small and concentrated, effective pieces, rather than as a few monolithic chunks. It corresponds in part to the point about the use of less code and less background information in conversation. When feedback allows speakers to reduce redundancy, it follows naturally that more, smaller exchanges lead to less 'dead weight'. To put it differently, the frequent exchanges make the interacting parts more closely adapted to each other.

Another important point is that both parties make *crucial* contributions to the conversation, and in some sense on an equal level, so that neither party is above or in control of the other. For example, this would be the case in conversation if there were a clear and rigid question–answer structure, where the speaker controls the direction of discourse, and the listener's contributions were subordinate to and regulated by the speaker's actions.

But there is also a derived, second sense of interaction, the one used for example when experimental psychologists speak of "interaction effects". With this, they mean that they have not been able to isolate a single, simple cause for a certain effect, but multiple ones that also influence or *interact* with each other, so that it is impossible to establish a simple causal relation.

In this case, interaction means approximately "a complex relation". This is the extended sense of interactive cognition, saying that the relation between cognition, action, and the world is not as simple as traditional theories have it. This relation cannot be reduced into a simple causal one, from cognition to action to the world. A linear relation is definitely too simplistic, but also a circular model is too restrictive. In interaction, transitions may come from anywhere and go anywhere, at any point. What really happens is determined by the contents and circumstances of each case, not by some general organizational scheme.

This complex relation also means that cognition and action cannot be reduced to two distinct phenomena. In my use, action retains more of its everyday sense, like "activity", or doing something, rather than mechanical motor behavior.

So at the core of interactive cognition is a process where cognition and action, or knowing and doing, are closely tied together, so as to realize the tight interaction between mind and world. The resulting process is so tightly integrated that it cannot be broken down into well-defined components with simple relations between them. Terms like "doing" and "knowing" can only emphasize and contrast particular aspects of this integrated whole, they do not correspond to distinct sub-elements.

Including action and world in cognitive explanations

Traditional cognitive theory is simple and therefore can explain simple things like experimental tasks and well-defined problems, but when more complicated cognitive domains are considered, typically more realistic tasks such as design, then a more complex relation between cognition and action (and so forth) is required for explaining what is going on. In realistic cases, cognition, action, and the world interact with each other in intricate ways that cannot be made to fit into conventional explanations, because they sacrifice too much detail in achieving their simplicity, leaving out too many aspects of what they are supposed to explain.

As a consequence, intramental cognition is a simple theory that can explain simple things, but to explain realistic cognitive phenomena we need a theory where the world itself is included in the explanation. One reason why intramental theory can be simpler is that it needs to make no references to the world. Thereby, for example, both action and perception can be left out of its explanations of cognitive phenomena.

In-the-world explanations, on the other hand, are not restricted to entities in the mind only. The world is not replaced by a problem space including only carefully chosen aspects of the world, and where these selected aspects have also been carefully coded so that a simple search algorithm will reach the right solution.

Instead, the interactive explanation includes such aspects of the world as artifacts and their properties. Thereby it can explain more complex phenomena, but it also requires the theory to explain how the head and the world can work together. It is a theory of how the mind *interacts* with the world and the things therein, and how individual and world jointly determine cognitive activity.

Interaction is the best way of using the resources of both mind and world to their fullest, just as conversation can draw upon the resources of both speaker *and* listener, unlike writing. The difference is that the writer *cannot* involve the reader, while the 'literary' model of cognition has *chosen* to leave out the world, as it has been held as an advantage to keep the world separate from cognition:

> Perception and motor behavior are assumed to take place in additional processing systems *off stage*. Input arrives in working [memory], which thus acts as a *buffer* between the *unpredictable* stream of *environmental events* and the cognitive system. (Newell, Rosenbloom & Laird 1989, p. 117, my italics)

5. Interactive cognition

The theory to be presented in this chapter consists of four steps. They concern the advantages of interactive cognition over intramental theory. Each step can be seen as a layer which is made possible by the layers before it, capitalizing on them to successively add further advantages of the interactive mode:

- Firstly, the advantages of dealing directly with the world instead of a surrogate for it, as the conventional theories do.
- Secondly, the advantages added by action and interaction with the world.
- Thirdly, a fine-grained structure of interaction that maximizes the benefits of involving world and action.
- Fourthly, a set of "shortcuts" made possible by drawing on the specific conditions of a situation rather than the general information a surrogate can only provide.

Step 1: The rediscovery of the world

The first step is to give back to cognition the access to the external world that cognitive science revoked very early. Because it has since then been claimed that blocking out the world was done with good reason, I will also have to motivate why such access is useful. I will therefore argue why and how the world itself can be more useful to cognition than a copy of it. I apologize in advance that the points I will be making here will appear self-evident to many readers. Nonetheless, they have to be stated since they go counter to the conventional positions of cognitive science. The programmatic way in which traditional theory has not "forgotten", or "neglected", but explicitly *kept* the world out of cognitive theories, is reflected in the quotation in chapter 4 about the need to place a buffer around cognition to protect it from the "unpredictable environment".

The existence of mental representations is often motivated by their ability to work as substitutes or stand-ins, as mental models and in mental simulations. Mental models are attributed with capacities similar to computer simulations, enabling them to imitate a physical system or process by embodying laws, equations and principles that describe the workings of that system (Gentner & Stevens 1983, Johnson-Laird 1983):

> If the organism carries a "small-scale model" of external reality and of its own possible actions within its head, it is able to try out various alternatives, conclude which is the best of them, react to future situations before they arise, utilize the knowledge of past events in dealing with the present and future, and in every way to react in a much fuller, safer, and more competent manner to the emergencies which face it. (Craik 1943)

This has been the original official motivation for having cognition operate on a mental representation of the world, instead of directly on the world itself. In particular, this is held to enable cognition to replicate and simulate events in the environment that go beyond the here and now of its immediate surroundings.

Still, the usefulness, and use, of mental models has not been restricted to events that are not directly available to the individual. The classical cognitive science approach to visual perception, to mention but one example, has always had the objective of producing a viewer-independent, complete three-dimensional model of the visible environment (Johnson-Laird 1989, Marr 1982). The reason for this is straightforward: Intramental theories *require* mental representations to work; their using them is not a matter of preference. For example, problem solving requires that the environment be encoded in a problem space, to be able to deal with it at all.

So we have two approaches to keeping cognition informed about and in synch with the world around it: by running an intramental simulation that shadows the events in the environment, as the traditional view does for one reason or another, and by checking the world in itself on a regular basis, not using any intermediate at all, which is what the interactive view proposes. What is the difference between the two? If I claim that checking with the original is simpler and better, intuition is inclined to agree. But since the opposite has been claimed so vigorously, the difference must be examined more closely.

Consider, for the sake of comparison, two different methods in ship navigation for determining the position of one's own ship, where this is done by simulation and measurement, respectively. One is known as *dead reckoning*. It is based on inferring the position of the ship by taking a known position and adding to it the ship's movement from that point. In principle this is an easy and computationally elegant way of determining position, based on simple mathematics. The movement may be computed from the speed and course of the ship and the elapsed time.

In reality, however, a ship is affected by external factors that make dead reckoning very difficult: wind, sea current, and so forth. These could be measured and taken into account, of course, but in practice it cannot be done with sufficient accuracy. And there are other sources of error, e.g. in determining the ship's own course and speed.

The other method, *position fixing*, uses references to elements outside the ship to determine its position. This method has been documented at length by Hutchins (cf. Hutchins 1995). In the specific case he describes, the bearings toward three landmarks are determined and plotted as lines on a map, and the ship's position is where the three lines cross and make a (hopefully) small triangle. This method is repeated at an interval of usually three minutes.

The specific ships that Hutchins describes are helicopter carriers of the so-called "amphibious fleet" (1995, pp. 7–9, 21). Ships of this kind are navigated by position fixing, not dead reckoning, at least when the ship is near hazards (within 8 kilometers) and therefore must be closely coordinated with its surroundings. This in spite of all the technical instruments and sophisticated navigation technology available, and even though all the scientific knowledge accumulated to this day could be used for constructing a model for dead reckoning the ship's movements and position.

Why is dead reckoning not accurate enough, and clearly inferior to position fixing? The purpose of determining the ship's position is to locate it relative to its surroundings, in particular to potential hazards and to where it ought to be or go. The reason why dead reckoning is worse than position fixing is that it is done by simulation based on a model or representation of the actual situation. The problem with simulations and models is that the world doesn't allow itself to be replicated accurately enough. It is simply too complex to be modeled with any precision. When textbooks use the laws of physics to model physical events, this is always done for idealized situations, where many greatly simplifying assumptions are made. Real measurements don't give the calculated results. Even in "real" physics such simplifying assumptions are made to an extent that may surprise the uninitiated.

Hence, computations yield errors, and these deviations from the real thing accumulate with each successive step in the simulation.

Thus the error aggravates rapidly, since each new estimate is based on the previous one, which already was wrong, and so on. So even a good model will begin to drift after a few steps; this is why weather forecasts only work a few days ahead, and become less certain for each day forward. In position fixing this does not happen. First of all, the data come from the real thing, not a model that is approximate at best. This alone makes the fix much more dependable. Moreover, the deviation doesn't accumulate since each fix is determined separately and is not based on the previous ones.

So ship navigation has a bridgeful of sophisticated technology and the accumulated results of science since Euclid to back it up. Still dead reckoning is not accurate enough. The same thing holds for cognition. Intramental cognition dead-reckons the environment, while interactive cognition goes straight to the source, without buffers or models in between. And like position fixes using landmarks, and like a speaker who has her listener available in front of her, her information about the world is not based on an estimate of what the world ought to be like.

The reason why I bring this up is that navigation by dead reckoning faces the same problems as the classical view of how actions are selected, by planning that is. A classical mental plan consists of a sequence of steps, each associated with an action. Each time a step is taken, the world changes from the physical action associated with it, as well as for other reasons. Because actions are not performed as they are selected, they must be selected on the basis of a mental simulation which dead-reckons the state of the world at that point.

This method was used in the robot Shakey in the 60's, and with exactly these negative consequences, where the constant issue of trying to keep an internal representation in synch with the environment became a major problem, as discussed more recently in e.g. Suchman (1987) and Dennett (1991).

This is the consequence of determining actions on the basis of a mental simulation, and this is how classical intramental planning has to be done. Because when the specification of action is separated from the execution of that action, then the consequences and the context of the future action must be simulated.

For example, if a communicated message is prepared in advance, as usually happens with written text, this could be done by "planning", by simulating the addressee's thinking after each sentence, to see if and how she will understand it. It doesn't have to be text, it could also be prepared speech. The risk of drifting further and further off the dead-reckoned course would still remain, however.

Philosophically, dead reckoning goes back to logical deduction from premises, whereas checking with the world is what one might call the pragmatic technique. In fact, some claim that "dead" is derived from "deduced" reckoning. Philosophy has always been concerned with how to reason properly. From syllogisms to formal logic, the aim has been to establish rules for making valid arguments and conclusions. This is the essence of the concept of rationality: the ideal, perfect way of reasoning; ideal thinking if you wish. Philosophers have always wanted to establish how you know what is right; how you know when you are right. This is part of what Dewey called "the quest for certainty" (1929).

So how do you know what is right? For practical purposes, you can simply check with the world to find out if you are right, if that is what you want. But philosophers are almost by definition not interested in practical matters, but are instead usually concerned with matters that cannot be settled by looking at the world. Metaphysics, Plato's world of ideas, ontology (What really exists?), epistemology (How can we know? What is knowledge? What is the relation between knowledge and the world?), What is truth? All of these are topics where the world can give us no answers. Other means are required, such as the principle of *reductio ad absurdum*: Everything that is contradictory and "logically impossible" must be false; something must be true if the opposite leads to a contradiction. In typical philosophical matters, principles like these are the *only* way of finding out, and the mother of all such principles is logical deduction: If something is certain, then other certainties can be deduced from it, step by step, each being perfectly logical, literally.

So logic makes perfect sense in the immaterial domains that philosophy is concerned with. The problem came when other sciences applied the same means to worldly ends. In cognitive science, it could be made to work for idealized domains like games and puzzles. These are also cases where you cannot check with the world to find out what is correct; you have to know the immaterial rules (cf. Zhang 1992, 1994) that are specified by a human and very similar to the rules of logical deduction.

But when applied to real cases, the problem of deduction or dead reckoning becomes a problem of constructing a model—a simulation—of the situation. This is why no dead reckoning model is exact or even good enough, and thus why logical deduction in immaterial domains does not transfer well to material ones—deduction works well in theory but not under authentic conditions. And if even the best model isn't good enough, then imagine how a model of a design problem that is very much under development would perform.

Instead real action must be grounded in feedback. This is what the three-minute fix cycle does. It is in this way analogous to moving–seeing–moving (sailing–fixing–sailing). By relying on feedback, all that remains background becomes harmless; every action is evaluated from its actual effects, not estimates. This is why designers, like Quist above, draw so that they can use their seeing to judge their own ideas, instead of trying to imagine what the consequence of a move is. They frequently see unintended consequences of their moves, and often these are desirable (Schön 1983). By appreciating the consequences of each step, an action sequence develops bit by bit as each step is performed, not in advance, and continual feedback from the world is used to stay on course. This is the pragmatic manner, which is synonymous with inquiry.

The pragmatic manner is a very simple way to find out. Logical deduction, dead reckoning, and so on are techniques that enable you to figure things out when the basic, simple way of finding out by checking is not available, as in metaphysics and so on, just as a writer has to compensate for not having her addressee in front of her. The rational ideal has made the mistake of regarding deduction from premises as the fundamental procedure for finding things out, not a compensatory technique for circumstances beyond the ordinary. It is thus like a literary model, in having been applied also to situations where the special, limiting conditions that motivate it do not apply.

Step 2: Manipulating the world—doing for the sake of knowing

The problems with a pure analysis phase have already been discussed, but the conversation analogy can give an additional angle on this issue: When the reader isn't there at the time of writing, the writer definitely can't ask a question and expect any answer. So analyzing the problem before writing the text seems rational; under these circumstances even the division into separate phases seems to make sense: first think, then write; first analyze the recipient, then design the outline, then write the text.

But when a speaker has her listener available right then and there as she speaks, why should she be content with passively using her perception for input, and use speech only to output her message? Then she would have to do with whatever feedback the listener was kind enough to give her. The second step in the interactive model concerns employing action for a cognitive purpose, analogous to the use of speech in conversation. The speaker producing the message corresponds to the first or productive purpose of action, from chapter 3—she can sign on speech to also induce feedback and to direct what kind of feedback she gets and when—this corresponds to the second, cognitive, or inquiring purpose.

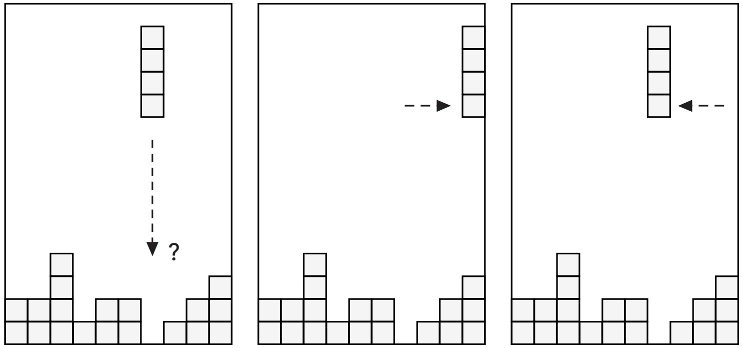


Figure 5.1 Is the Tetris brick lined up correctly? Finding out, and then ensuring that it is, by first moving it three steps to the wall and then three steps back.

It is a cognitive purpose since it contributes to performing the cognitive task; its effect achieves what a mental simulation would. If she gives speech this second purpose, then the feedback she gets will be richer, more to the point, and much more useful to her in designing her message.

When Kirsh & Maglio studied subjects playing the video game Tetris, they found that their subjects made moves—physical actions—that could only be explained as serving this second purpose (1992, 1994). In Tetris, bricks in different shapes fall down onto the playing field, and the objective of the game is to build the growing pile on the ground so that it does not fill the playing field all the way up, which will end the game (cf. figure 5.1). The way to do this is to fill the assembling horizontal rows completely; such a full row will disappear, and this is how the player is to keep the pile low. Bricks fall down one at a time, and the player can move the falling brick left and right and also rotate it, so as to fill the bottom pile evenly without leaving holes. Kirsh & Maglio were able to demonstrate that players make certain moves that actually bring the pieces *away* from their goal position. That is, these moves definitely do not serve the productive purpose of action, which is to reach the goal of placing the brick; they are even counterproductive in this sense. Instead these moves have a demonstrably cognitive purpose:

> … certain cognitive and perceptual problems are more quickly, easily, and reliably solved by performing actions in the world rather than by performing computational actions in the head alone. We have found that some translations [i.e. left and right] and rotations are best understood as *using the world to improve cognition*. (Kirsh & Maglio 1992, my italics)

Their first example of such actions is when the player was to fill a gap say three steps from the wall. Instead of relying on a mental visualization to determine whether the falling brick is lined up correctly, players moved the brick to the wall—that is, away from where it will go—and then moved it back by pressing the proper key one, two, three times. In this way the player could ensure that the brick was over the right position, three steps from the wall (also figure 5.1).

In a second example, players rotated pieces that were not yet completely visible, to determine what kind of brick was coming. This enables them to decide earlier where it should go (figure 5.2). This action cannot have a productive purpose since it is made before the player has decided where to put it. Players also rotated pieces more often the more ambiguous they were (*ibid.*). Kirsh & Maglio also showed that rotating a piece on the screen is much faster than rotating it mentally, and that it is likely that pieces indeed are rotated so as to see where they would fit.

These are two examples of actions that have cognitive purposes, that is, where the individual interacts with the world to perform cognitive functions which traditionally have been attributed to purely intramental processes—in particular mental simulation and prediction. They follow the pattern of a speaker who evokes responses from her listener, and thereby uses speech as a kind of action that serves a cognitive purpose, and that makes intramental cognition both superfluous and inferior.

Dewey elegantly described this as "doing for the sake of knowing". As he also noted, this is a phenomenon that we can find everywhere in ordinary everyday activities, if we only look for them:

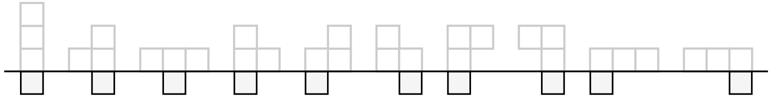


Figure 5.2 By rotating a falling brick that is not yet completely visible, you uncover hidden parts to faster determine its full shape.

> The rudimentary prototype of experimental doing for the sake of knowing is found in ordinary procedures. When we are trying to make out the nature of a confused and unfamiliar object, we perform various acts with a view to establishing a new relationship to it, such as will bring to light qualities which will aid in understanding it. We turn it over, bring it into a better light, rattle and shake it, thump, push and press it, and so on. The object as it is experienced prior to the introduction of these changes baffles us; the intent of these acts is to make changes which will elicit some previously unperceived qualities, and by varying conditions of perception shake loose some property which as it stands blinds or misleads us. (Dewey 1929, p. 87)

It is also through action that our pragmatic knowing is used and put to a test, and it is in this way that we can see whether it serves its purpose or not.

Exploration

Inquiring action can be divided into two kinds: *exploration* and *experimentation*. I will begin with exploration. The quotation from Dewey captures its everyday meaning: "to make changes which will elicit some previously unperceived qualities, and by varying conditions of perception shake loose some property which as it stands blinds or misleads us". This is why active manipulation betters pure analysis. If you are trying to understand an object, aspects that are not immediately apparent come out if you manipulate it; instead of just passively watching the object, you act upon it to see what happens, such as by rotating a falling Tetris brick to see its obscured parts. Exploration is a fundamental and very common aspect of inquiry, regardless of domain:

> This is much of what an infant does when he explores the world around him, what an artist does when he juxtaposes colors to see what effect they make, and what a newcomer does when he wanders around a strange neighborhood. It is also what a scientist does when he first encounters and probes a strange substance to see how it will respond. (Schön 1983, p. 145)

When design specifications are in themselves very incomplete and leave much unsaid, so that just a pure analysis of them would be insufficient, that would correspond to a speaker who merely takes and uses whatever information the listener will give her. But to elaborate the given requirements, and go beyond what is obvious or explicit in them, designers do just like speakers—they use exploration to evoke what the specifications do not mention, and to make out what consequences follow from them. Guindon *et al.* have provided a detailed account of exploratory practice in software design: "[Subject] P8 explicitly acknowledges the need for exploring the problem environment to achieve a good understanding of the requirements before seeking a solution." (Guindon, Krasner & Curtis 1987, p. 69)

Adelson & Soloway (1985) noted that an experienced designer working in an otherwise familiar domain used exploration when he came to an unfamiliar part of the problem. Also Guindon showed that exploration is associated with understanding unfamiliar material. A good understanding allows for systematic work since it gives the designer a map to follow, while little experience requires her to explore without a sense of direction:

> [Subject P6] clearly has better design schemas [i.e. understanding] for the communication sub-problem than for the scheduling problem. He successively refines his solution for the communication sub-problem while he performs much more exploratory design for the scheduling problem. By exploratory design, we mean design with many mental simulations of the problem environment and mental simulations of tentative solutions unguided by a plan. (Guindon *et al.* 1987, p. 68)

The authors state that the designer's main method of exploration is through *simulation* of the eventual context that the artifact will enter into. Simulation is prototypical as a physical design activity where the actions involved have an essentially cognitive, inquiring purpose. Guindon has described the use of simulations in great detail, in her case in the design of an elevator control system (e.g. 1990b). All of her subjects simulated scenarios of how the elevators would be used. Firstly, these served to help them understand the requirements and the problem domain, and thereby to infer requirements and to generate solutions. Secondly, they used simulations to explore the solutions they were developing, to find inconsistencies, incompleteness, or bugs. In other words, to understand the solutions they had developed themselves. Still, both kinds concerned the same setting, only with or without having the design in place. Both kinds of simulation occurred throughout the design process (*ibid.*).

> I'm going to imagine one elevator and a few scenarios. Say there's a request from floor 2 to 4. If there is a lift going to 2 on its way up, then stop the lift at 2, open the doors, ... If there is a lift going down from 5 to 1, the lift does not stop at 2 ... What if you press up at the floor, but once in the lift, you press a down button. ... So there's definitively the need for a queue of lift requests for each lift, separate from the floor requests. ... Maybe the floor requests could be handled by a completely separate system from the lift requests. (p. 287)

In this protocol excerpt, the designer imagines a couple of scenarios. By simulating the consequences of certain actions she works out what happens step by step, and this eventually leads her to discover a situation that was not immediately apparent:

> By simulating a Lift scenario, the designer realizes that a user may press a floor button to go in one direction, but once inside the lift, may press a lift button to go in another direction. This test case was not mentioned in the problem statement, yet it is critical for the design of a good control algorithm. (p. 288)

Thus this single scenario developed her understanding of the design problem by singly generating a test case, a requirement, and a solution, at the same time. In another example the designer explicitly links her simulations to the need for understanding a certain aspect of the problem, namely scheduling (p. 286):

> I'm not sure I understand about scheduling. I'll draw two elevators with a few floors. ... For each lift, I have, say, four buttons that are illuminated or not. And for each lift I also have to know the floor and the direction. Say Lift 1 is at floor 4 and there are

> requests to go down to floors 3 and 2. … The floors don't move, the lifts move. It strikes me that I haven't considered enough this idea of having lifts between floors. I'm going to handle that.

Here, to understand the issue better, the designer creates a scenario that involves scheduling, and simulates what would happen. She elaborates one particular case to see what the system should do. In this way she discovers a test case for evaluating solutions, plus an additional requirement.

These simulations enable the designers to explore their problems and thus go beyond the limited information that is directly available to them. In this way they develop their understanding by tinkering with what she has, not merely analyzing the given specifications. Exploration is then a typical case of how action can have an expressly cognitive function.

Experimentation

Experimentation is more powerful than exploration. In fact, exploration can be seen as a limited version of experimentation that lacks certain elements of the "full" process. By making experiments you physically *test your ideas* in the world, instead of trying to figure out in your head what will happen.

If exploratory speech is when a speaker probes the other party by asking her questions, then experimentation in conversation is when a speaker is not just asking questions but actively tries to make her point, and uses the feedback she gets to find out whether her speech is working or not. She has some notion of what she wants to say and how to say it, and by actually saying it she conducts an experiment and can evaluate the outcome, that is, the listener's response. She can use the response she gets (the "consequences" of her talk) to see whether her idea works.

In design, experimentation is the main method for testing and working out ideas. Quist's sketching episode can be regarded as an experiment that he makes to work out his proposal and to test the consequences that come with it. After stating his framing he starts to draw to see where it leads and whether it will work. A string of moves draw out the possibilities contained in his idea, and he speaks and draws simultaneously: "…that could then be the bridge, which might generate an upper level … we could have as much as 5-foot intervals … The section through here could be one of nooks". Through this drawing experiment he tests his idea and is able to make something good out of it; it generates possibilities rather than problems. The experiment thereby "confirms" his framing in some sense.

As Quist then goes on to advise Petra on how to proceed, he instructs her to experiment to find her way, in the same way that he just did (Schön 1983):

> Q: Well, that either happens here or here, and you'll have to investigate which way it should or can go.

That is, she should make experiments to test her ideas. A little later, he is also explicit about what designers' experimentation consists in—working out your ideas using paper and pencil, *trying them out* by drawing:

> Q: Now the calibration of this becomes important. You just have to draw and draw and try out different grids.

Here Quist stresses the cognitive, i.e. non-productive, purpose of drawing. He emphasizes the process of drawing, not the product. She should "draw and draw", and "try out". This is drawing as an investigation; as an inquiry.

Another episode, even though Schön uses it for a different purpose, gives an even clearer illustration of how central experimentation is to design, and of how closely related it is to sketching. Here, Northover is the coach and Judith is the student presenting her problem (1987, pp. 127–132):

> *Judith*: I haven't decided yet whether it's going to be sited right here or right here—I have the feeling it's going to be here and I'm going to make it level.
>
> *Northover*: Do you have this to a larger scale somewhere?

When Judith describes her design, Northover asks for a scale drawing. He needs to *see* her placements to be able to judge them. Making such scale drawings is seen as an essential drawing experiment (*ibid*., p. 127). The reason is that the relation between building and site is very important, as previously discussed. By drawing the building to scale, in its location and on a site plan, the designer can examine this relation in detail and work out a proper placement. Above, for example, Quist expresses this importance when he asked Petra if the drawing she has made is to scale, and she answered yes. But Judith answers:

J: Not right now, no. But it works as far as southern orientation —being far enough from here so I don't get drainage problems, being near enough to this flat area so I can set up playgrounds. ...

N: So you don't have it on a site plan at all!

J: No, that didn't seem necessary...

When Judith answers that she doesn't have a scale drawing, Northover expresses his astonishment over not only that she doesn't *have* one, but also since it implies that she hasn't even *made* one. She "feels" that this was not necessary, but to him it is crucial, since it means that she hasn't made the experiment of drawing the site plan to scale, which he sees as essential for evaluating her idea:

> Northover seems to be saying, "You are not designing at all. You are simply having 'ideas' and putting them down on paper. The moves you make have consequences that are testable, but you must draw to scale and in section in order to test them. The whole process of designing is lost to you because you will not do these things." (*ibid.*, p. 130)

Judith continues by describing "a ramp which spirals up". Northover then asks for a floor plan. Again, she says that she thought it was not necessary. She proposes to put "art and cafeteria" on the main level, and she asks him what he thinks. He answers, "That is possible, I guess", and asks about level changes and circulation. Judith expresses her wish: "Most people will use the ramp".

Again, the same clash over drawing to work out ideas: Northover neither approves of nor rejects her proposal—because without having the idea and its implications worked out on a drawing, he simply cannot evaluate it. When he says, "That is possible, I guess", he is really saying, "It might well be a good idea, again it might not; it is just that I cannot tell whether it is without having the idea worked out for me. With anything less I can only guess." He then goes on to give her some constructive advice:

N: I think you have got to really discipline yourself to draw it up to scale and draw a section through it—let's just assume that these ramps do work, that access—if so, this ramp will cut off the views to and from the library.

Here Northover tells her that the drawings that she thought unnecessary are crucial—she has to "discipline" herself to do them, because they are drawing experiments that she must make, to test whether her ramp will work or not. Drawing is the way designers test their conceptions, to see in detail what they lead to, and to develop them from mere ideas to reliable concepts that have been tried and confirmed. Judith does not see this function in drawings. She "decides where it's going to be sited", she "has the feeling it's going to be here" and she "is going to make it level".

I bring up this particular dialogue because it is as if Judith holds the conventional cognitive view of drawing, since she doesn't draw to *work out* the consequences of her ideas and decisions; she draws only to *document her choices* for others to see them. The use of drawing as experimentation is what the conventional view has overlooked. It has regarded drawings as a medium for recording the end product of design, or at the most as an extension of long-term memory (Akin 1986, Goel 1995, Newell & Simon 1972). In this view drawing is not regarded as experimentation but as output or storage, an epiphenomenon of pure thought. Drawing adds nothing to a problem-solving process that is purely intramental.

> Judith simply has no idea what Northover means by drawing, conceived as a process of trying out design moves and discovering their consequences and implications. ... it is clear that she sees drawing not as thought-experimenting but as a way of presenting ideas (Schön 1987, p. 130)

Again the conventional cognitive theories can be said to correspond to novices' beliefs and working styles, and the poorer results that this leads to, whereas the techniques of seasoned designers reflect the interactive approach.

Why are experiments (and simulation) in the physical world superior to models and simulations in the head? The reason is that you want to find out both what you *can* figure out and what you *can't* figure out, i.e. what you cannot simulate mentally. That is, you want to know also about the effects of your actions that you cannot predict or foresee.

Dewey's original purpose with his theory of inquiry was to explain the role of experimentation in science. It was the method of experimentation that had made physicists capable of their monumental advances ever since the Enlightenment. It was also the ancient Greeks' resorting to mere speculation that had made their progress in the natural sciences so marginal, especially in relation to their contributions in immaterial domains—philosophy and so forth—by the use of the same method. Indeed, thought experiments were the single method used in Aristotelian physics. (Or is it the other way round—that philosophy hasn't improved much since those days because of its reliance on such procedures?) In some sense, the non-pragmatist theories of knowledge and scientific reasoning thus remain at the stage of Aristotelian physics.

This shows the limitations of simulation in the head—for that is what a thought experiment is—compared to interactive experimentation in the world. If an idea is tested on a mental model, then the test reveals only those consequences of the idea that are accounted for *by the model*, i.e. the aspects that you have incorporated into it. The remainder is left out. This is why thought experiments never disconfirmed the Aristotelian idea that heavier objects fall faster than lighter ones. An experiment in the world can reveal to you both unanticipated consequences of your idea, and also limitations to your *model*, since surprising consequences indicate shortcomings in the model. Neither of these could come out of testing ideas in a purely intramental way.

This is just as position fixing does not depend on the navigator's understanding of winds, currents, and so on, and thereby is the superior navigational method. Dead reckoning, on the other hand, is only as good as the best available model of these physical phenomena, which obviously isn't good enough yet, and quite probably never will be. And however much it is improved, all the fuss involved can only aspire to eventually become *as good as* the method of simply looking out to see where you are.

Here, I am not trying to say that we are bad at mental models and mental simulation—but that these limitations give cognition good reason for not being intramental; for not using mental models and simulation in the first place. This holds for my argument in general: my purpose is to give cognition reason for not being intramental at all, not to show that it is intramental albeit badly so.

Step 3: Fine-grained interactive structure; economy and efficiency

Having concluded that checking with the world has advantages over intramental cognition, and that manipulating the world brings even further advantages, it is a logical extension that these means should be used in the best possible way. That is to say, if checking with the world is good, then check often and make use of the information you get as *much* and as *often* as possible. And if manipulating the world improves your knowing even further, then you ought to use this technique to its full potential, too. The result would be fine-grained pieces of activity, a continuous attention to feedback that replaces complex pre-planned actions, and simpler and smaller actions that both generate feedback and attend to and adjust to it.

Why is this better? Remember dead reckoning: it starts from a known position, but the shortcomings of prediction yield an accumulating error that makes the computed position deviate more and more from the actual position. Position fixing instead reestablishes accuracy each time a fix is made. So the more often you make fixes, the more often can you make the proper adjustments to your course, and the less will your measured position drift away from the actual one.

This leads to an organization of cognition that is radically different from when cognition is strictly separated from everything else, as happens in traditional theories. The difference in strategy is clear in the following brief example. Here, one person is trying to help another locate a building on a map (Brennan 1990, cited in Clark 1996):

> *no to your right, no over by the quad, right there yah right there*

There are a number of interesting points in this brief and deceptively simple example. First of all, it shows that in normal face-to-face conversation you *can get feedback* on whether what you say works as it should, and that this leads to higher certainty, allowing the speaker to say "yah right there".

This example also shows the predicted *fine-grained* pieces of activity. Instead of a precise description in a complex sentence with embedded clauses and elaborate structure, which would be characteristic of written language, there are several short and very simple segments, which is typical of ordinary conversation. The segments also display a *high sensitivity toward feedback*. Because the speaker continuously attends to visual feedback, she can formulate each segment on the basis of what the addressee is doing. Thus, the speech co-evolves in parallel with the addressee's activity. Thereby it is possible to make out what happened just from what the speaker said.

The speaker can achieve this high sensitivity because fine-grained action has *local control*. When action is specified in a large chunk before it is performed in an equally large chunk, there is no way to make use of feedback. Fine-grained segments, on the other hand, enable local control since they are specified not in advance but as they are performed. Thereby there is a way for feedback to enter into the process.

Conversation is known to have local control (Sacks *et al.* 1974). Among different kinds of spoken language, conversation is on the most-flexible end of the scale, it is completely locally managed, and comes in segments much like those of the present example. Between each segment there is a slot, or what Sacks *et al.* call a "transition-relevance place", where others may give feedback or take the floor, or the current speaker may continue and thereby make a multiple-segment turn, as the speaker did in this example. The size, content, and speaker allocation of each turn are thereby managed locally.

By this single and sole control mechanism, turns at talk are assigned to participants locally, i.e. one turn at a time. More organized scenarios such as debates may instead for instance assign equal shares and ordered turns to speakers in advance. Local management is done in interaction between participants. A speaker may or may not stop voluntarily at a transition point, and others may attempt to initiate a turn at such a point, or they may remain silent, thus encouraging the speaker to continue. For these reasons a locally controlled system is maximally flexible and adaptive to circumstances (*ibid.*).

With fine-grainedness and local control, action is specified—or *designed,* as Clark & Wilkes-Gibbs (1986) call it—as it is being carried out, so it can be altered on the basis of what actually happens. In the example, when the speaker "designs" her instructions she can take advantage of what the addressee is doing right then—where she first puts her finger, just how much too far in some direction she moves it, the exact moment when it passes over the right spot, and so on. Thereby there is no need for the redundancy that would be necessary for a description that is designed in a separate phase, prior to being performed. If the speaker couldn't get feedback, she would have to use an expression that does not depend on how the addressee will use it to locate the spot on the map. Typically, such a phase would first have to refer by name to something that is easily found, then specify each next step from there, with name and direction, plus for example the number of streets to go past, and so on. The need for all this redundancy would lead to a complex expression that is both much longer and much harder to design—all of this made necessary by having separated the specification of the action from its performance.

The same advantage of conversation over writing has been demonstrated for definite references (Clark 1992, Clark & Wilkes-Gibbs 1986, Krauss & Weinheimer 1964, 1966). Over repeated trials, speakers use feedback to make references shorter, as in the following example (Clark & Wilkes-Gibbs 1986):

1. All right, the next one looks like a person who's ice skating, except they're sticking two arms out in front.
2. Um, the next one's the person ice skating that has two arms?
3. The fourth one is the person ice skating, with two arms.
4. The next one's the ice skater.
5. The fourth one's the ice skater.
6. The ice skater.

If the "grains" of talk are sentences as in this example, then clearly speech becomes fine-grained compared to the writing-like sentence 1: The long reference with several parts in different kinds of relations to each other—part–whole, (i.e. person–arms), activity/function (skating), location (in front of) and so on—is replaced by short and simple ones, both in structure and in syntax.

Also note how the speaker uses rising intonation in line 1 as a question-like prompt for feedback, at the point where she presumably would begin to try to make the initial reference more economical. Krauss & Weinheimer (1966) showed that speakers made their references shorter only when they received feedback: "By monitoring the listener's responses to his encoding, the speaker was able to decrease the number of words needed to code a given figure without running a great risk of being overly cryptic and confusing to the listener." (p. 344) That is, to be more concise without running a risk of not being explicit and elaborate enough.

The authors also examined feedback of two different kinds. When feedback was given concurrently ("mmm", "aha" and so on) there was a much greater difference than when the addressee gave feedback only after the reference had been completed. This supports the argument that when you can get feedback, then the more fine-grained interaction is, the better it works.

Incremental, approximating sequence

In Brennan's (1990) direction-giving example, the segments of talk are formulated on the basis of what is going on at the time when they are spoken. What is said in one segment has effects in the world, and following segments make use of these effects. For example, if the previous segments had not made the addressee move in the right direction, the segments "no over by the quad" and "right there" would not have worked. In other words, these segments make use of their particular context. So the specification of each segment is based on what happens in the world at the same time—speech and action co-evolve in parallel—and the inquiring effects of the spoken segments make the world a particularly valuable resource when the speaker formulates the segments that follow.

As the example shows, changing to a finer grain means that not only action, but also the *specification* of action is broken up into a sequence of smaller pieces. Instead of being planned separately and in advance, specification is done concurrently and together with performance, in a sequence of steps distributed over the whole course of action. The specification of each step builds on the outcome of previous steps, in an incremental fashion.

This fundamental change in procedure means that the method of specification becomes experimental; it becomes an inquiry. In the example from Brennan, the segments of speech serve as *experiments*: the speaker tries an instruction that might work, it has certain effects on the addressee (she moves the pointer). The speaker can then evaluate the outcome of this "experimental" instruction and adapt the following segments accordingly. This experimental procedure gives a role in inquiry to action, as doing for the sake of knowing. The fine-grained, incremental form of her speech is necessary for giving her spoken actions this second, inquiring role.

The traditional theories of action only consider its *productive* effects. The consequence is that according to this theory, actions are only specified so as to produce the desired result; to have a complete productive effect so that they "get it right". Their productive effects are their *only* effects. If specification is incremental, as I am proposing, its objective instead becomes to conduct an inquiry that produces a good outcome *in the end,* not at once. This gives action a different role which is not only productive. The purpose of action is no longer just to give the right result, especially not right away. Instead action is specified to also serve its inquiring purpose, as doing for the sake of knowing. It should manipulate the world to evoke feedback, and serve to test the knowing behind it. So cognitive effort should not mainly be spent on figuring out one large and complex action that produces the right result on the first attempt, but on performing an incremental sequence of simpler actions that concludes in a good result; the result will in this way be firmly grounded through the experimental nature of this procedure.

As a consequence, individual actions become *approximations* instead of perfect, once-and-for-all actions, because they are not specified so as to produce the desired result each on its own. Not approximation in the mathematical sense, but in the sense of being rough and unfinished while moving inquiry forward, because their inquiring effects enable upcoming actions to work better. Thereby the individual actions shouldn't be judged as being correct or wrong, but as parts of a larger sequence that leads to a successful result. That is, individual actions should be evaluated for how they work as parts of an inquiry, not from whether they produce the desired result at once. For example, "no to the right", shouldn't be classified as insufficiently specified, or as a "correction" of an earlier imperfect instruction, but as a concise and thereby efficient part of an incremental sequence. This sequence succeeds rapidly and with little energy, by using an interactive strategy that depends on entities in the external world for its success. In the example that I have been using, the "external entities" are the addressee, her actions, and the objects that she manipulates by these actions.

This approximating model makes one particularly important prediction: that the action that makes up the first attempt at something should be very different from how monolithic pre-specification would have it. There, the first action should be the only one necessary. It should be precise, well-conceived, and have only a productive purpose, that is, only serve to bring about the desired result. Here, in contrast, the productive effect of the first action can be minimal. Instead, the initial objective should be to get an inquiry off the ground. It should be a "starter", exactly what it does is not all-important, not too much cognitive effort should be spent on it. Getting the outcome right, the productive purpose, would become more important further on when the inquiring purpose is fulfilled. It has then worked out how to produce the desired result.

The previous example of direction-giving seems to bear this out. No segment appears to be the result of deep thought—their common purpose rather seems to be to make the other party do the work of finding the right spot! That is, to first get her to start searching, and then merely give her a push in the right direction when needed. The first segment, "no to your right", at first sight does not even look like a starting segment, but on closer consideration it makes perfect sense as such if only the addressee has her finger on the map, or otherwise appears to focus on a particular part of it. So if we only accept it as a first segment, then no doubt the thought behind it was minimal. Instead it fits perfectly into the description of a starter; its obvious motive is to make the addressee begin moving her finger. It is hardly an imperfectly constructed exhaustive description that is repaired afterward.

Clark & Wilkes-Gibbs (1986) also found what I call starters, as well as other inquiring techniques, to be characteristic of collaborative talk: Of the eight kinds of techniques for making references that they discuss, six have an inquiring function: "in three examples [speakers] deliberately drew the addresses into the process; and in three they began by knowingly issuing a questionable or inadequate noun phrase" (p. 113). The inquiring function is clearly present in these cases, sometimes even dominating—the authors themselves conclude that they "do no more than initiate the process" (p. 122). For example, speakers used *try markers* (rising intonation) to bring in the other party, compare with sentence number 2 listed above. These serve as attempts to start an interactive approximating process. They also spoke noun phrases in multiple segments, each inviting affirmation and whose continuation depends on the response (*ibid.*).

Step 4: Pragmatism enables specificity and shortcuts

Shortcuts in inquiring function

In exploration, the inquiring function is explicit—exploration is doing *only* for the sake of knowing. In conversation, this corresponds to a plain question. In experimentation the inquiring function is instead usually *implicit*, as part of an action that also has a productive function. In other words, such an action has both a productive and an inquiring function at the same time. This productive function is typical of experiments in practice, whereas it is usually absent in science (Schön 1983, pp. 147f). A designer, unlike a scientist, understands a situation and changes it at the same time, she is conducting a *productive* inquiry.

The direction-giving example above is a demonstration of the advantage of inquiry and experimentation over intramental specification. All the segments have a productive function: the speaker has an initial idea about how to direct the other, and each of these short instructions serves to produce that result. But each segment is also an experiment (cf. Schön 1983): The speaker's idea of what to do is her "hypothesis". Each segment of speech puts an aspect of her idea into effect by making the addressee move her finger (or hold it still in the final case), and so it tests her hypothesis as a scientific experiment would. It thereby spells out the consequences of the "hypothesis", and you can judge whether the hypothesis works or not. Moreover, it also shows *how* it works and how it does *not* work; each segment—each one a small experiment—lets the speaker develop and adapt her idea further, as the experiment advances her understanding of her situation.

Here we can see how specification and performance proceed together and in parallel: At no point does specification advance far ahead of production. At each step, specification builds on the outcome of the previous step, in an incremental fashion. The first segment adjusts for direction, in the second one the speaker decides to point out the quad, probably from the addressee's finger movements, and then the test that is implicit in "no by the quad" shows that she has succeeded; "yah right there".

This also shows the advantage of experiment over exploration. It is hard to imagine how the speaker could use exploration—"Where do you think it is?" or "What places do you know?"—it seems contrived and hardly very efficient. Questions (and exploration) are too vague, they do not test any specific idea about a solution. Neither do they have any productive effect.

Because actions are also implicit tests of the idea behind them, a designer doesn't have to be explicitly concerned with experimentation; with making tests or evaluating their outcome. Instead, she can rely on the *breakdown* mechanism (this concept originated with Heidegger 1927/1962). For this reason, as long as things go well, as long as there is no trouble, she can simply keep on doing what she does and focus on that without worrying about what might happen, and so forth. Breakdown only occurs when something goes wrong, only then does the designer become aware of the test.

This principle is also at work in conversation, where a speaker ordinarily does not explicitly test whether the audience is understanding what she says. Instead she simply speaks, giving *opportunities* for feedback, and its absence works to confirm that the audience is following her. In our direction-giving example for instance, the spaces between segments serve as such opportunities for indicating trouble that are not taken. Were this not the case, the addressee having trouble following the directions would signal this.

In this way, what a speaker says implicitly tests whether it is understandable, and the absence of feedback works as an implicit confirmation that the addressee is understanding what is said. She does not even notice any implicit tests, constantly thinking, "Since they're not protesting, what I'm saying must make sense to them. Good, then I can continue". Instead, of course, she is busy enough speaking.

For this reason, experimentation can be largely transparent even though it very effectively tests every action and reveals any problematic consequences. It is completely transparent as long as no troubles arise. This enables experimentation to be quite effortless: testing an idea can simply consist of attempting to carry it out. If this succeeds, then you have also conducted an implicit experiment that proves the idea to work, but you have at the same time produced the very result you wanted; experimental verification comes at no extra charge. And you experience your own activity as only being concerned with producing the result.

Perhaps the inquiring function of action has been overlooked precisely because tests are implicit in this way. We do not experience or intuitively see our own actions as tests, even though we experience breakdowns and adjust to them, which shows that actions do work as implicit tests. We do not recognize their inquiring function, only their effect on the world. This would then also be the reason why traditional theory of action similarly only considers its *productive* effects. The consequence is that according to such theory, actions are only specified (planned) so as to produce the desired result; to have a complete productive effect so that they "get it right".

Shortcuts in productive function

Because each attempt also works as an implicit test, speakers can go beyond the first level of laziness, which is to avoid working hard on specifying attempts, and reach a second level. This is what I will call *optimism* for want of a less intentionalistic term. Again consider these minimalist lines from a casual dinner conversation (Tannen 1984):

A: Do you *read?*
(1.0)
B: Do I *read?*
(0.5)
A: Do you read *books?*

Even with this highly condensed speech, we may safely conclude that A is trying to ask B whether he reads fiction books in his spare time. By starting with an exceptionally brief "Do you *read?*", A is acting like an optimistic high jumper who chooses to enter into a competition at a very high level. If she can make the jump, she will have saved a great deal of effort by skipping all the jumps on lower levels that less optimistic competitors must spend effort on clearing. But if the optimistic high jumper doesn't clear her entry level, she will end up last on the scoreboard, registered as not having cleared any level at all. For an optimistic speaker, on the other hand, the situation looks much more promising: In Tannen's example, A's first optimistic attempt fails, but in the second one she can back down to a lower level and be more elaborate. And as we see, this time it works, even though the second attempt still must be regarded as being very optimistic, as it remains very terse.

But also B must be considered very optimistic. "Do I *read?*" does not reveal much about how B fails to understand what A said, and B doesn't try to make A's repair easier, by for example offering a interpretation (such as "Do you mean…?"). Still, A manages to repair this problem at once.

The advantage of optimism becomes clear if you consider the problem of adapting a message to the addressee's background knowledge, and of being explicit enough while not being redundant. As discussed earlier, a writer who is separated from her audience must provide ample context so that they will understand her, by adding redundancy to her core message. By being optimistic, a speaker skips a maximum amount of redundancy. But her optimistic attempt also works as an implicit test, and as an experiment with the addressee's understanding: If there is a problem, the addressee has several kinds of response available to indicate the problem source in her feedback (Schegloff, Jefferson & Sacks 1977). In this way an optimistic attempt is a powerful way to induce precise feedback which will become helpful if a second attempt is necessary. In the light of Garfinkel's (1967) demonstration that exhaustive descriptions of finite length cannot be made, optimism seems to be a particularly valuable way to determine a proper amount of context.

So, someone might object, it appears that optimism supports the concept of starters and incremental approximation. But couldn't it be that these techniques are used only when they are warranted, and that standard, exhaustive pre-specification still is the standard procedure, although speakers avoid using it when they can? No, because when a speaker appears to sense that her specification will be problematic, she doesn't spend more effort on specifying the first attempt. Instead she does the opposite, and makes the inquiring and incremental nature of speech explicit by marking her attempt as tentative, and by also encouraging the other party to collaborate (taken from Sacks & Schegloff 1979):

A: ...well I was the only one other then then the uhm tch *Fords?*
uh Mrs. Holmes Ford?
You know uh [the the cellist?
B: [Oh yes. She's she's the cellist

A: Yes. Well she and her husband were there.

(The bracket denotes both speakers talking simultaneously.) This example demonstrates almost every phenomenon mentioned so far in steps 3 and 4: A's hesitation ("then then the um tch") displays her trouble and marks her attempt as problematic. It is followed by her starter, *Fords?* This is clearly an optimistic attempt in the present sense, and it also has a rising intonation as a try marker requesting explicit confirmation. (This reverses the role of absent feedback. It comes to mean continuing *trouble*.) In the absence of feedback, A makes two more approximating attempts. Both use the same pattern as the first attempt: hesitation-marker–attempt–try-marker: "uh Mrs. Holmes Ford?" and "You know uh the the cellist?" Here, "Mrs. Holmes Ford" is a second attempt and a more explicit version of "Fords", which shows that this is an optimistic sequence.

So in this case, A seems to be aware of her problem in advance, but she doesn't spend more effort on conceiving her first attempt. On the contrary, a second look at this example even suggests that in the event of trouble, speakers in effect spend *less* effort on getting the first attempt right. This would then mean that they instead emphasize the starter's inquiring function, to invite the addressee to contribute to the interactive process. In other words, it seems as if speakers try to exploit the advantages of the interactive procedure to their fullest when they need it most, and that they therefore *emphasize* the inquiring and interactive aspects of conversation when they sense trouble. This tendency is even more pronounced in the following excerpt (Jefferson 1973, p. 59):

A: I heard you were at the beach yesterday. What's her name,
oh you know, the tall redhead that lives across the street
from Larry? The one who drove him to work the day his
car [was–
B: [Oh *Gina!*

A: Yeah Gina. She said she saw you at the beach yesterday.

Here, both "What's her name" and "oh you know" are minimally informative, so in a sense A appears to be maximally optimistic. It is as if she believes that B can tell who she is thinking of from the context alone. With "oh you know", she even repeats her appeal for help before trying herself. In whichever way a listener interprets what A says, her words serve to start an interactive procedure if not much else. So it seems that she prefers this interactive procedure to pre-specification.

So actions are approximate: they are not specified to be complete and perfect. They are also optimistic: they are specified with little cognitive effort and make shortcuts. Instead, the emphasis in action is on the span of the whole procedure, rather than its start. What about the end of an approximating sequence? When does it reach its conclusion? Also the end of an incremental sequence comes without much commotion. Generally speaking, it ends when it has reached its practical purpose. However, the end comes quietly because there is no explicit evaluation of success, just as tests are not explicit, neither their evaluations. Instead, following the same principle, success is the absence of breakdown. If you try to perform an action and you succeed, then you are finished—automatically and implicitly, without concluding so, or testing whether you are. The end of one action or action sequence is particularly implicit and invisible when it is immediately followed by another action, and this is the normal case under realistic circumstances.

As a result, actions are specified *viably,* not correctly or perfectly. Through attempts, more or less optimistic ones, and trouble that spurs further attempts, specification proceeds until it *works*. That is, until there is no more obstruction, so that nothing remains to be done. It stops because there is nothing more to do, not because an explicit evaluation function has been satisfied (or satisficed).

There are several points were we can find the viability principle at work. There is the phenomenon from conversation that you do not point out or repair a speaker's mistake if it doesn't present any problem; if you can figure out what she meant, or if it is not very important to the purpose at hand, then you simply do not object (Schegloff *et al.* 1977, p. 380). If it ain't broke, don't fix it.

Schober and Clark (1989) also found that speakers and addressees in cooperation settle on referential expressions that are not "correct" or objective, or even *intelligible* to a third person. Instead, they "exploit adventitious commonalities" and settle for the first perspective that makes sense to both. That is, they select the first viable candidate expression. If a third person cannot make sense of it, then she is, as they put it, out of luck. The resulting references are specific and local to them, not "objective" universal descriptions. If that had been the case, an outsider would have no trouble understanding it.

All shortcuts are made possible by viability principle

I have adopted the term *viability* from von Glasersfeld (1982). It is the same explanatory principle as in evolutionary theory and the principle of natural selection. Natural selection is not the survival of the fittest, it is the non-survival of the non-fit. In the same way, only the non-fit actions are improved on. It is not the correct and most fit actions that are prepared and then performed. With this I want to stress that the individual actions in incremental sequences do not approximate a correct action increasingly well, in the same way as evolution does not proceed toward the ideal creature.

Instead, as Bateson (1967) has stated, this kind of explanation is *negative*, because it is not the production of the effective, but the elimination of the ineffective. The principle of implicit tests and breakdown, which I introduced above, is of the same, negative kind.

Breakdown only occurs because further progress is impossible; actions that are good enough, even though still sub-optimal, pass without notice. So actions are viably specified even though there is no evaluation criterion; or more correctly *because* there is no evaluation. This is just as there is no selection mechanism in evolution either—evolution is only the non-survival of the non-fit.

Making the world "cooperate"

When a process is interactive, then it is almost by definition determined by all interacting forces. In Brennan's direction-giving example, even though the addressee doesn't say anything, if we want to explain why the speaker says what she does, it is clear that this depends to a great extent on factors apart from herself. When she tells the addressee, "no over by the quad", or "right there ya right there", these statements are appropriate because of external events happening at the same time. Because of the incremental nature of approximating sequences such as this one, the meaning of an utterance depends on external events, which in turn have been brought about by previous utterances. In this case, "right there" relates to a particular finger movement, which itself was the effect of "no to the left" and "no over by the quad", etc. Similarly, on the first "right there" the finger presumably stops, which makes the repetition "ya right there" appropriate.

Hence, the unfolding actions of each party are highly dependent on the actions of the other party, which in turn depend on your own previous actions. Each individual's actions, here her utterances, cannot have been determined only by her own mental processes, as traditional cognitive explanations would have it. That is to say, we cannot explain why the speaker said exactly this without including elements outside herself in the explanation. An exhaustive description produced in the manner of the literary model, on the other hand, would not have involved external factors in this way. So this interactive manner of specifying the location to the addressee gives these elements a role in determining what is actually said.

As a result, the "passive" external world is promoted to an important role in this process. It must thereby be recognized as having a fundamental and systematic influence on the direction of the process, an influence of the same order as that of the actor herself. Thereby the process is determined interactively by individual and world together. However, saying that the world is a part of cognitive processes appears very alien, both to the cognitive scientist and the nonprofessional.

Still, there has been a similar situation in the study of conversation. Both intuitively and scientifically, speech production has traditionally been thought of as one party, the speaker, producing output and the recipient merely receiving speech—the "literary" model of speech production, based upon how written language is produced. However, studies of talk have shown that speaking in conversation is a fundamentally interactive and collaborative process, i.e. where what is said is determined jointly by speaker and addressee in close cooperation (for example Clark & Wilkes-Gibbs 1986, Goodwin 1979, Sacks *et al.* 1974, Schegloff *et al.* 1977). And this is of course what I have been taking advantage of in comparing interaction to conversation, and intramental cognition to writing. By analogy, traditional cognitive theory is based on a "literary model of action", where actions are the output of the final step; the literary model of speech production is a direct application to language production of the intention-to-action schema, and the rational model of action as a whole (compare with chapter 1).

Pulling the world into the cognitive process is what makes the interactive and incremental strategy so effective. The brief directions ("by the quad", etc.) accomplish their purpose single-handedly, almost lazily, by offloading effort onto outside factors, doing with minimal effort and deliberation what would have required significant energy for traditional models to achieve. The small, simple segments of speech have an effect in the world, and it is these effects that allow the speaker to use much less effort in achieving her end. This is partly due to the transfer of work from one party to another, but also partly because, to an outside observer, this strategy decreases the *total amount* of work done by all involved parties, by eliminating the redundancy that is necessary to compensate for separation, as detailed above. Hence, it works more effectively not only for the speaker, but also as a whole.

The brief, deictic "to the right", "by the quad", and "right there" that are spoken would not work in the literary model. They are so brief because they can just "point" at circumstances in the situation that are available. In effect, these brief expressions specify the location jointly with the situation; they determine it in interaction with each other. The simultaneous presence of both—the absence of separation—allows the spoken specification to become *interdependent*

with the world it is referring to. Separation instead would require a written description to be independent of circumstances. It would have to be self-contained, because it couldn't involve elements that might not be present.

Steps 3 and 4 in relation to sketching

In steps 3 and 4 I have mainly based the presentation on examples from conversation. It can in one way be regarded as a closer cognitive analysis of Schön's (1983) concept of "design as conversation" with the working materials, which is reflected in the dialectical structure of sketching. And it is this dialectical and highly interactive structure of sketching that the analyses in steps 3 and 4 concern.

Step 3 demonstrates the value of the moving–seeing–moving structure of sketching, whose pattern adheres very closely to the fine-grained, interactive structure that is described there. Sketching is made up of very small and simple incremental steps, which yield local control and high sensitivity to feedback. This, in turn, makes sketching into a highly fluid and efficient process, which supports the open-ended and conceptual nature of the design work which sketching is typically used for.

The concepts from step 4, in particular optimism, improve on this by enabling the designer to move forward and test ideas very rapidly. By merely starting to work on a solution by sketching, she can make substantial shortcuts, and there is no need to think first, and then draw the solution. She can just start to work out her ideas, and simply back up and be more careful or elaborate if it turns out she has been too optimistic.

Moreover, the incremental approximating sequence elaborates on the developing dimension from chapter 3. This is the structure which results when you cannot separate the specification of an action from its performance. Instead both of these processes run in parallel throughout the process. The concept of starter serves to point out that in such a process, the quality of the first solution attempts should mainly be regarded as starting the process, which will eventually lead to a satisfactory solution.

In this chapter, I have begun to answer some of the questions that were posed in chapter 4, with a model where activity and working materials are true parts of the cognitive process in itself, for example. As seen in the following quotation, even though the importance of sketching has been recognized, a dichotomy is still made between sketching and thinking:

> Thinking is one of the most notoriously intractable parts of psychology since the thought process is not easily observed. ... The designer, however, has never resembled Rodin's "Thinker" who sits in solitary meditation, but has in contrast always externalized his thoughts, not only as an end-product in the form of a design, but as an integral part of the process itself in the form of drawings and sketches. (Lawson 1980, p. 96)

6. Making the world a part of cognition

6.1 Inquiring materials

The topic of the present chapter can be put in two different ways: speaking generally, it is concerned with the role of the world in interactive cognition. On a more immediate level, the argument concerns this issue as it materializes in design, in the cognitive role of physical working materials in designers' activities.

The role of action in cognition was the topic of chapters 4 and 5. The heart of the argument there was that cognition is not organized around a mind working in isolation, but to carry out cognitive tasks through making the most of mind, action, and world working in concert. For this reason, cognition is organized differently, and thus works differently, than if it were purely intramental. The result is that when these three parts work *together*, performance is superior to that of an intramentally organized cognition.

The present chapter will carry this argument on to analyze the role of the physical world in this scheme: how can the world have a role in cognition, and what exactly is this role—what does the world contribute to cognition? A short tentative answer is that it makes effective interactive cognition possible; this is the role of physical working materials in design. Interactive cognition relies on mind, action and world working together; its superior performance depends on the immediate presence of those physical materials that it is concerned with.

Such a short answer is not very informative, however, and a more substantial answer requires a deeper analysis. For this, I will begin by returning to the topic of sketching, although this time it is the *sketches* that are in focus; the *material* rather than the activity. From the previous discussion of sketching, it should intuitively be somewhat clear that sketches have a cognitive purpose, just like the sketching that produces them. That is to say, they are used by cognition and contribute something crucial to it.

However, the stakes on the relation between cognition and world are quite high, and so many things have been said about it, that a convincing case needs to do much more than appeal to intuition.

Figure 6.1 Two kinds of sketch for a magazine cover, with two different levels of precision and refinement. Note the relative sizes of each kind.

The substance will this time be found in the sketches themselves, and in other kinds of working material that designers use.

I will start out with the notion of *inquiring materials*: working materials with a cognitive purpose. This idea is analogous to that of inquiring action, which has been discussed at length in the previous chapters. Much of the argument here parallels and builds on the points that were made there.

The analogy lies in the following: the artifacts we normally associate with design are the *products* of the design process. Hence we have a productive purpose of materials, just as we normally think of the productive effects of action. If however a material is created in the design process, not as an end product, but rather to serve the inquiry that the design process is, then it has *a second, inquiring purpose*. Again, this parallels the inquiring purpose of action. An "inquiring material" then, like and inquiring action, does not function as an *end* product of design, but as a *means for* the inquiry that design is. Sketches have this very purpose, and are therefore the first kind of inquiring material examined in this chapter.

6.2 Sketches

While the material about sketching came from architecture, the best evidence for the cognitive purpose of sketches comes from graphic design:

> Right from the earliest stages of tackling a problem, designers' thinking is mediated by the sketches or visible notes that they make to familiarize themselves with the material they are manipulating. On paper, these notes may be very rough to start with: possibly just thumbnail sketches that indicate the structural relationships between elements of a document without focusing in detail on any particular elements. (Black 1990, p. 284)

This is the use of sketching that we recognize from chapter 4: for graphic designers as much as architects, sketching is the way in which they work on a problem. In the very early stage that Black describes here, designers make sketches to "familiarize themselves" with their problem. As the work proceeds, sketches are used in many other ways, too.

Still, a statement such as this is not enough for determining that sketches have a cognitive role. The case will be much stronger if it can be shown that sketches do not serve any productive purpose at all.

If so, then they are created for a cognitive purpose only, and therefore qualify as a pure inquiring material. The evidence for this lies in the properties of sketches themselves. They do not look like production drawings—the kind of drawing that is made as the end product of design, and which is passed on to other people—they look much more informal and unfinished, and thereby lack the precision required for serving any productive purpose (cf. figure 6.1).

A second reason is the following: Why would sketches be created from the very beginning of design, when just about everything is likely to change before you arrive at the final product? This would surely be a waste of effort. Hence, their purpose must be to serve the inquiry that creates them. And since they are created so early on, they have to serve this purpose right from the start.

This is an essentially negative argument—they can't be anything but inquiring materials. However, the fundamental reason behind this roughness is positive: their being unfinished is what makes sketches suitable for their inquiring purpose. But what is more, different kinds of sketches are given different properties, to fit their different purposes. For example:

> Rather like language, these drafts have *different levels of precision and formality* (inflexions) depending on whether designers are using them for feedback to themselves, or for communicating ideas to colleagues and clients. (*ibid.*)

Thumbnails

Hence, the format of sketches in general is adapted for different purposes. Sometimes, such an adapted sketch format is a category important enough to get its own name. One format used by graphic designers is the *thumbnail* (figures 6.1, 6.2). An introductory textbook in graphic design gives the following description; I have emphasized the expressions that indicate their cognitive function:

> Thumbnails are *idea sketches*; they are visual evidence of *the thinking, searching, sorting process that brings out solutions.* ... Thumbnails are usually small because they are *meant to be fast and undetailed.* ... Fill a sheet of paper with evenly spaced small rectangles and then fill them with ideas. Never reject an idea; just sketch it in and go on. Work through the idea with your pencil from every perspective you can imagine. Then try taking one idea and doing several variations on it. ... *Be as neat and precise as is necessary* to show the relationships between elements and their general shapes. (Arntson 1993, pp. 5–6, my italics)

Figure 6.2 Thumbnails (dimensions somewhat reduced).

Arntson here details the process that creates thumbnails. What makes them noteworthy, and a category of their own, is their purpose—they are the solution to a very common problem in graphic design. As Black noted above, thumbnails are used right from the beginning. Graphic designers use them to *explore* the problem and to become familiar with it (chapter 5). This exploratory use is what thumbnails are adapted to. First, as Arntson states, they are used for generating many ideas with several variations, and therefore they should be fast to draw. This is why they are made small, if not literally quite as small as thumbnails. Second, exploration does not involve testing ("rejecting") ideas, or going into great detail. Therefore they "may be very rough to start with", and should only "indicate the structural relationships", according to Black. The small size also helps to keep them rough; you cannot add much detail on this scale.

In this way, it is clear that the properties of thumbnails as a physical working material are thoroughly adapted to both their purpose and the process of drawing them. They are small because they are meant to be fast and undetailed, and they should only be as neat and precise as is necessary to show the relationships between elements and their general shapes. Thereby, thumbnails are given properties that enable them to support, even enhance, a rather specific use, the exploration of a graphic design problem. This close relation between their physical properties and their purpose and use activity is a very significant fact that applies to all kinds of inquiring material.

Arntson also describes how thumbnails are created. This is a working procedure well suited for its exploratory purpose, spanning a wide area of possibilities without heading in any specific direction or searching for a particular goal. The thumbnails' physical characteristics have a major impact on the form of this exploratory sketching activity, and these characteristics are also essential in making it effective.

The working material has certain characteristics that produce a highly adaptive and very effective working procedure. This enables the designer to focus on those aspects of her problem that she is presently concerned with—the general questions that are important early on—without having to work out the full details of design concepts that will later turn out not to work and which will therefore be abandoned. These questions can instead be postponed until later. Hence the properties of the material and the format of the activity are closely adapted to their purpose. In this way, physical materials can have an impact on cognition that is both specific and substantial. This is also good evidence that thumbnails are an inquiring material.

Roughs

In addition to thumbnails, Arntson gives a corresponding description of *roughs* (cf. figures 6.1, 6.3). If thumbnails are exploratory, then roughs are best characterized as *experimental* (cf. chapter 5):

> Once the range of ideas has been fully explored, select the best two or three thumbnails for refinement. ... *The purpose is to test whether the idea still works on a larger scale*. Take this opportunity to work out small problem areas that could not be dealt with or foreseen at the thumbnail stage. (1993, pp. 6–7, my italics)

Figure 6.3 Roughs (dimensions greatly reduced, cf. figure 6.1).

Hence in roughs there is once again a close parallel between purpose, process, and the properties of the material. Roughs are used when the best ideas are to be taken beyond their initial conceptions at the thumbnail stage, and be worked toward a final result. Arntson lists the main purpose of these roughs as *testing* the best ideas; a switch from thumbnails to roughs is thereby also a shift from exploration to experiment. She also notes two other functions: firstly, that such a test implies further refinement as well, and secondly, that you now work out problems that did not appear in the thumbnail format (cf. figure 6.1).

To fulfill these objectives, the properties of this particular kind of sketch are adjusted accordingly: Whereas the minimal size and imprecision of a thumbnail are well matched to its uncommitted stage of work, the rough goes one step away from this by increasing both scale and precision. Thus it becomes possible to evaluate the ideas more thoroughly. Again, the increase in detail is partly a consequence of the larger format.

Conversely, one could say that a larger format is necessary if you want to work at a finer level of detail. The thumbnail format simply does not measure up to any serious testing of an idea; this is why Arntson specifies that the test should be done "on a larger scale". Also because of the format, roughs are the right medium for further refinement, as well as for dealing with issues that "could not be dealt with or foreseen at the thumbnail stage".

As a consequence, roughs also require more work. That is why you have to select only a few thumbnails to work out in greater detail. This is however entirely as it should be, it is not a fault of thumbnails that they cannot do this. Thumbnails are used not in spite of their small size and lack of detail, but *because of them*; the same holds for roughs—each kind suits work at a certain stage of the design process.

The greater amount of work is rather a way of adapting procedure to purpose. With roughs the working process changes, from the rapid creation of many simple thumbnails into longer episodes of more detailed work, concentrating on just a few alternative designs.

This leads to an important point: It is a mistake to measure any kind of inquiring material by how different it is from the final product. These differences are not shortcomings, they are the very essence and *raison d'etre* of design materials that are created for a cognitive purpose.

These two kinds of inquiring material, thumbnails and roughs, can be said to directly correspond to the two kinds of inquiring action introduced in chapter 5, experiment and explore. While both kinds of inquiring action are doing for the sake of knowing, exploratory action is rather vague and probing by nature, in that it lacks a specific direction. Experimental action is more specific, in that is also works as a *test* of the action itself, and it is thereby also potentially much more rewarding.

Transferring this distinction to inquiring materials, thumbnails and roughs are two kinds of "material for the sake of knowing", i.e. with a cognitive purpose. Thumbnails are exploratory sketches, used to generate visual ideas and explore the problem, but not to *test* the ideas. This interpretation finds support in Arntson's advice to never reject an idea at this stage of work. A rough is in contrast an experimental sketch which is used just as such a test, and accordingly its characteristics are adapted to serve this purpose best.

So sketches are an inquiring material whose properties and work processes are adapted both to how they are used and what they are used for. But they are not merely fitted to their inquiring function in general, the adaptation goes even further, as the cases of thumbnails and roughs show. These two kinds of sketches have different purposes, and they are therefore given different properties to better fit their specific functions (cf. figure 6.1).

6.3 Situating strategies

With this initial analysis completed, we may now return to the theoretical issue of the role of the world in cognition, which is that when the objects of your concern are physically present in front of you, they enable interactive cognition to work at its best. This much I have already stated, but before any specific consequences can be derived and examined, as I am about to do, the general principle must be made into something much more tangible. To begin with, what are the objects that are of concern to the designer?

An obvious object of concern is the artifact that is being designed, but it is not the only one. The designer's concern with function requires her to consider more than the isolated object. Even though this has probably always been recognized in the design literature (e.g. Alexander 1964), the need for pointing this out may lie in the everyday sense of "design", as it appears in "designer clothes", "I like the design of your watch", and so on. For some strange reason, design in the everyday sense seems to be as remote from function, usefulness and other practical concerns as it could ever be. In fact, when something is referred to as design, it ironically enough always seems to refer to those aspects that have no purpose, but which exist only as decoration or embellishment.

In any case, exactly what the designer must take into account may not be obvious. To design in the more precise sense discussed here, function is a central concern. This means that the artifact that is produced is not the genuine objective of design work. In terms of function, the artifact is not an end, but the *means* by which the designer can achieve her real end. This actual objective is to effect certain changes on a particular situation, which is typically quite complex. The artifact's *function* equals the role it will play in this changed, future situation. This role will usually be quite complex, as the artifact will have effects in many dimensions: social, organizational and others as much as the physical domain (cf. Brown & Duguid 1994, Ehn & Kyng 1991, Greenbaum & Kyng 1991a, b). The designer's job is to ensure that the resulting function is indeed the one that she desires. Accordingly, her concerns will span a large share of what might be called the functional situation, covering all the elements and dimensions that are relevant.

Computer science often separates usability from functionality, but I find this distinction largely redundant. How an artifact works together with the people who use or operate it is a fundamental part of its function, not something apart from it. To my mind, such a separation only reflects how much computer science has historically been concerned with issues where human involvement is marginal. The emphasis that has been given to usability was necessary to make up for this negligence, but to maintain the distinction here would yield unnecessary complications. Moreover, the distinction is irrelevant to most design disciplines; only a small fraction of all artifacts operate without human involvement. Hence, to keep things simple, I will not separate use from function.

The designer re-creates the future situation of use

Taken on its own, the claim that physical presence enables interactive cognition appears somewhat empty. However, there is a special circumstance that applies to design, and herein lies the twist: The designer's inquiry concerns the situation that I have just described. But this situation is not present to the designer, and therefore not available to her interactive cognitive process. Not only are the designer's concerns remote from her; the functional situation is located in the future, and hence does not even exist yet. Therefore, the given basic provisions do not enable the designer to nurture a healthy, interactive cognitive process; she will have to settle for something less.

The artifact itself, for example, will only come to exist as a result of design, when the work has been done. The designer's concerns are remote even when an existing situation is to be changed. While studying the existing situation is useful, it only takes you so far—after all, the future conditions are what really matters. Unless the design itself is trivial, the existing situation will be changed in non-trivial

ways. Therefore, steps still have to be taken to get at the non-existing, future conditions.

The remote conditions that the designer needs to understand can be characterized collectively as the *future situation of use*. Not only the artifact by itself, but also far from anything and everything in the world. For instance, design can be described as an inquiry into this future situation of use.

Designers have a very distinctive and unique way of making up for this problem: There is a range of design techniques—with names such as *sketching, prototyping, mock-ups, scenarios, storyboards, simulation,* and *user testing,* among others—that vary greatly in their surface characteristics, but still use the same strategy to *enable the designer to get at the future situation of use*. Quite simply, these techniques re-create the various parts of this situation that do not yet exist. To make interactive cognition work well, the designer has to create her own working materials; before the world can become a part of cognition, the designer has to create it. Therefore, I will collectively refer to these design techniques as *situating strategies*. They serve to make the world a part of cognition.

This provides a unified explanation of these techniques that are all widely used in design practice, and that are well recognized to be of great importance, but for which we heretofore have lacked a deeper theoretical understanding. This is, I believe, much because they do not adhere to the prevailing kinds of explanation (cf. chapter 1). Nevertheless, once the need for re-creating the physical world has been formulated in this way, it orders all these apparently diverse phenomena under one simple explanatory principle. As it will turn out, their diversity comes with a purpose: the future situation of use is a concept that applies to situations whose conditions may be very different from one another. Each strategy re-creates one aspect that may occur in a future situation of use. It is the diversity and large number of situating strategies that allow them to fill their common function across a wide range of conditions and purposes, and different kinds of design.

In this I am restricting the discussion to materials that are produced as part of design, thereby excluding any ready-made materials that might have a cognitive purpose. This makes the case stronger for the world's role in cognition, while strengthening the criteria for what qualifies as evidence: If a designer can be shown in this manner to go out of her way to create materials that serve no productive purpose, then this makes a stronger case for their cognitive importance, than does merely *using* existing materials.

How the strategies are used

It can now be made clearer what role the world plays in cognition. It has been established that it is the re-created future situation of use that represents the "world" here. What is the re-created future situation used for? What kind of interactive cognition does it enable?

As this is best explained by example, I will keep this general introduction brief. Still, design is an inquiry into the future situation of use. The function of the situating strategies is that they enable this inquiry to be performed interactively, with the advantages that were described in chapter 5. This means that situating strategies *a*) always occur as part of an inquiring activity that *b*) makes use of what it re-creates: there is never such a process of re-creation that is not tied to an inquiring design activity. This connection means that the resulting activity is a single working process with two logical components: one is the strategy that re-creates the future situation of use, the other is the inquiry itself, which uses that which is re-created. However, these two components are so highly blended together that an analysis of such an activity cannot strictly tell them apart. This is among other things due to the fact that one action can have both an inquiring and a productive purpose at the same time. For instance, as Quist's demonstration of sketching showed, the activity of making the sketch is impossible to distinguish from the inquiry that makes use of it. At any rate, what is important for the present discussion is that without the situating strategy that produces the working material, the inquiry that uses this material would not be possible.

The general principle is that situating strategies serve to make interactive inquiry possible. Still, their function can be divided into two general categories. The first one is where the strategy re-creates the object of inquiry itself. The value of having a prototype of the eventual design itself in hand should be obvious, particularly compared to evaluating the design on the basis of written specifications. This also illustrates how the situating strategies can improve cognition, since it shows the difference between inquiring into an aspect with or without having a re-created version in hand. Holding a prototype allows the designer to inquire into the design interactively, rather than by intramental imagination of a design (a remote/nonpresent entity) that is merely described in writing.

The second function is when the re-created aspect is used in an inquiry into some *other* aspect, as support in a sense. For instance, it is easier to inquire into an artifact's interaction with users, if not only a prototype is made, but it is also placed in the hands of a representative user. Here, the reason why is not as straightforward. Since the function of everything (not just the artifact itself) involves interactions with other aspects, an inquiry into a given aspect must also look at these interactions. This interplay is easier to study if both of the involved aspects are made concrete, since this also makes the interaction between the two available to hands-on inquiry.

I should also add that all parts of the future situation of use are equally eligible to be the focus of inquiry. The designer is no less interested in inquiring into for example the user, than into the artifact itself. Prototypes may often be useful in the second role, as support for inquiring into the user and other aspects.

6.4 Prototypes

The probably most common situating strategy and working material used in design is the *prototype*: a lifelike model of the design-in-progress itself, made as the design is being developed, and that is equipped with some of its properties.

The classical kind of prototype is the one used in industrial design (figure 6.4): three-dimensional models, made of some easily manipulable material, and often in life size if practical. The use of this kind of physical, hands-on model is not restricted to industrial design, but also occurs in other disciplines, including more recent ones such as interaction design. There are also many other kinds of prototypes that are used in many different design disciplines.

The purpose of the present section is to go deeper into the question of what makes a good inquiring material through an analysis of prototypes. What characteristics make a material good for cognition in general, and for design in particular? This elaborates on the analysis of thumbnails and roughs, and the observations that these materials have their respective properties adapted to their specific inquiring purposes.

From their commonness, the uses of prototypes in design are very diverse. The power drill prototypes in figure 6.4 were created to explore and evaluate the possibilities for improving handling comfort and reducing noise, leading to the famous eight-hour grip, a breakthrough in industrial design ergonomics. Earlier models had been

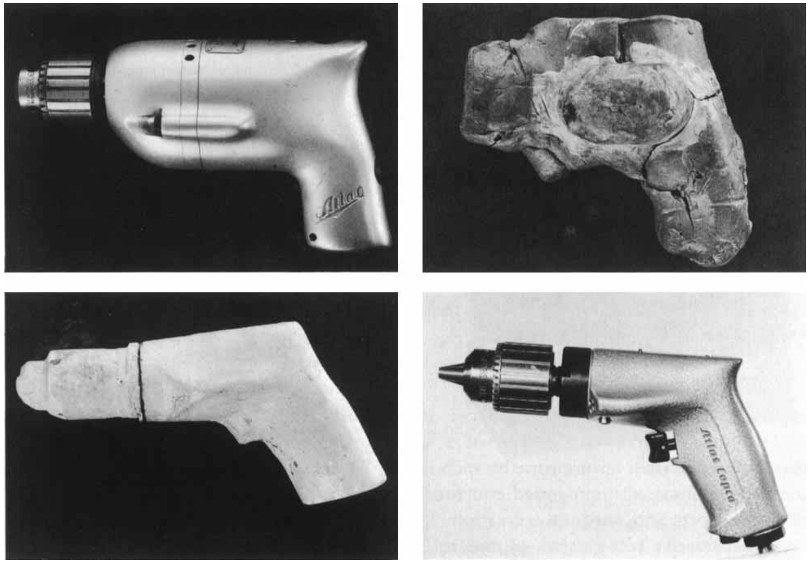


Figure 6.4 Industrial design prototypes and the final products.

inspired by science-fiction space-guns of the time (seriously!), and their styling thereby hampered practical use (Heskett 1980).

The realness of lifelike prototypes makes them particularly easy to relate to, in particular with respect to how they will function in their eventual use, rather than as isolated objects detached from a realistic setting. Prototypes give the designer a concrete, tangible model that she can relate to physically and practically, rather than intellectually. The same advantages make them suitable for getting feedback from potential users and customers, a function that is at least as important.

However, in some cases, models of a future product may be created purely as showpieces; to impress clients, potential customers, and the media. Often, they are even necessary for persuading the management and higher executives of your own company (Shrage 1996). This kind of use is what we see in the "concept cars" that automobile makers often present at trade shows. In the eyes of the general public, prototypes of this kind may arguably dominate the image of what an industrial design prototype is: Albeit unrepresentative as prototypes, this kind simply has the greatest impact, and the visual appeal to attract attention. In reality, such a showpiece does not share the purpose of prototypes as discussed here, but qualifies as such merely by the dictionary definition, as a precursor of things to come, or as the first in a series.

This is also the reason why images of actual working materials are so hard to find in the design literature: they are not considered pretty enough. For illustrations, these are often "polished" or completely redrawn by an illustrator to become presentable (all drawings from the Quist and Petra episode belong to this category). Sometimes this is even done contrary to the author's wishes (cf. Alexander 1971). And when they make it to print, to "look good" they are often presented as book illustrations normally are, with a distinctness and clarity that removes their transient character, which is their very essence. But their lack of presentability often causes the designer to throw them away herself, too, making them hard to document and study.

Perhaps even more clearly than with sketches, the partial and *deliberately* unfinished nature of prototypes shows that they must have a cognitive purpose. Why else would a designer go out of her way to create a model of the design long before the product is finished? When she knows that it is certain to be changed, when she uses a special prototyping material, and when she also knows that it can only be given some of the properties of the final design? In cases such as the power drill prototypes (figure 6.4), the resemblance is even minimal. Therefore, it is clear that prototypes are not built for a productive purpose.

Just as with sketches, the properties I have just enumerated are not problems but indeed desired features, as they make prototypes simpler than the real artifact. This makes them suitable for use in inquiry, for instance to test the design as it is being created and refined. Classical design methods always had an evaluation phase at the end of the design process. The problem is, there is little use in evaluating the design when you are done with it, as it is too late to use the test results. Testing should be done when the results are still useful. With a prototype, you do not have to wait until the design is completed, you can perform tests very early and use the results to inform design. This kind of testing is also known as *formative evaluation* (Hix & Hartson 1993).

As a prototype is a model of the design itself, working on the design may simply consist of developing a prototype which *is* the design that the product will be based on. In this manner, no other design specification is used at all, and design as a whole becomes an entirely physical, interactive working procedure that involves no abstract, logical specifications or other means of that kind. Historically, this has arguably been much more common: until the 16th century, historians have it, the artifact was built directly; drawings, prototypes, or other intermediate forms were hardly used at all (see e.g. Cooley 1988, Herbert 1993). Such hands-on work undoubtedly qualifies as an interactive working procedure in every respect. At that time, design by drawing first began to emerge. Even after that, abstract non-resembling design specifications, written or other, have been rare, and even then being used mostly as instruments for business and legal aspects of design projects (Lawson 1980). Hence, design as work directly on prototypes probably prevails still today.

Also like graphic sketches, the materials used for prototypes are chosen to match the fluidity of design: balsa wood and other soft, workable woods, Styrofoam and other similar materials are common (cf. figure 6.4). The same applies also to prototypes of less physical, seemingly more "intellectual" prototyping work, such as in software or interaction design.

User interface prototypes in paper vs. software

In the discussion of thumbnails and roughs above, I made an initial analysis of the desirable properties of an inquiring material. For prototypes, this analysis can be made much more comprehensive. It will be based on an analysis by Rettig (1994) where he compares user interface prototypes built in two different media, paper and software, arguing that paper is a superior medium. Being a practitioner's comparison of two different kinds of prototype that share the same purpose, and which points out the advantages of one kind over the other, this provides good insights into which qualities are desirable in a prototype.

Rettig's (1994) discussion of user interface prototypes—i.e. for the screen contents and visible behavior of a computer application—compares prototypes made in paper and software; two alternatives that are quite different from one another. Prototypes built in software are computer-based, and their graphic screen displays and interaction are potentially indistinguishable from those of a real program. Accordingly these are known as "hi-fi" prototypes.

The other kind is so-called *paper prototypes*: user interface mock-ups basically created on plain paper (*ibid.*). Typically being drawn by hand and laid out on a table, paper prototypes give a very informal and non-technical impression, and they are impossible to mistake for a real user interface on a computer screen. For these reasons, they are also known as "lo-fi" prototypes.

These two alternatives represent two almost diametrically opposing approaches to the same prototyping problem, and this makes the prototyping options available to a designer stand out very clearly, with their strengths and weaknesses. The following comparison will show that the properties named as *desirable* by Rettig precisely the ones that make a prototype work well *as an inquiring material*. This also serves as very good evidence for prototypes having a primarily cognitive purpose.

The merits of a prototype divide into two major kinds, building vs. using. The obvious kind of merit is the extent to which they enable interactive cognition to work well. One might also call this their efficiency as inquiring materials. I will be referring to this aspect as *using* or *testing* (i.e. inquiring into) the prototype. The second kind could be called their efficiency as situating strategies, since e.g. prototypes must be created before they can serve as the basis for inquiry. For this reason, a dimension to consider is how easy they are to build, before they can be used: how simple they are to create and also to modify. I will call this *building* or *working with* the prototype itself. I will begin with this second dimension, as it is conceptually simpler to explain.

Workability and relevance

When Rettig (1994) compares paper and software prototypes, he makes the following remarks on their workability (p. 22):

> *Hi-fi prototypes take too long to build.*
> ... Paper prototypes, on the other hand, are extremely fast to develop and the technique is very easy to learn.

The point is that paper prototypes are simple to build. Hi-fi prototypes are much more difficult to work with. They are almost as complicated to make as the final program itself, since they, too, are made in software (*ibid.*):

> Even with the coolest of high-level tools, building a prototype is still essentially a programming exercise...

Software is a much more difficult medium to work in than paper and pencil. Whereas you can just start making a paper prototype off the top of your head, a prototype in software is a design project of its own, that requires special consideration to avoid potential difficulties. A prototyping tool does help a great deal, but you still need to know what it can do and what its limitations are (Ehn & Kyng 1991). And without a tool of this kind you are facing a full-scale programming project.

Software prototyping tools also require specialized skills, whereas Rettig points out that paper and pencil require no training above kindergarten level. He adds on a more serious note that a paper prototype allows you to focus entirely on the features of the design itself, without having to also figure out how to realize these features in the prototyping application or programming language.

> To make a broad generalization, interface designers spend 95% of their time thinking about the design and only 5% thinking about the mechanics of the tool. Software based tools, no matter how well executed, reverse this ratio.

The paper medium is also very robust, whereas hi-fi prototypes are sensitive to even the very smallest technical details, like all software. Minimal problems can bring the prototype down completely, or keep it from running at all. Fixing problems and making modifications become time-consuming tasks. Rettig notes that a “bug” in a paper prototype often can be fixed as it is being tested on a user, with just a small interruption, and only using pencil or eraser:

> ... we all know how hard it can be to get all the bugs out of a program. On the other hand, I often see teams correcting “bugs” in a paper prototype while the test is in progress.

Relevance

These points are all related to how easy the two kinds of prototype are to work with, for the designer who is building a prototype. Several aspects are mentioned: how quickly you can build it, how hard the technique is to master, how much you can concentrate on the design rather than figuring out how to do what you want, how robust they are, and how easily they can be changed.

I will suggest that all of these points derive from one basic principle, *relevance.* The whole idea of building prototypes is that they are like the final design in many respects, but not in all. In particular, they lack those features of the final product that are unimportant for the purpose that the prototype is used for. This obviously saves work—under good circumstances, a *lot* of work.

This can be expressed as a principle of relevance: A good prototype serves its purpose as a basis for inquiry and interactive cognition, while being simple to create. This means that it should have the properties required for its purpose, and as few other properties as possible. It also means that relevance is always relative to just what exactly a prototype will be used for; this determines what properties it will need to have. Hence a good inquiring material is the product of a successful trade-off.

How can relevance account for all the above questions? Take the obvious ones first: Relevance means leaving out unimportant aspects. This saves work for the designer, it makes the prototype easier both to build and to modify. It holds for all prototypes, too; however, it holds to different degrees in different media: working in paper allows you to leave out more than in software prototypes.

This is because you can often leave out *detail* in a prototype, and thereby save work because for its purpose the prototype does not need the details. It is often sufficient to make rough, preliminary versions of its component parts, compared to the full detail of these parts as they will appear in the final product. In other words, you gain relevance by being able to leave out unimportant detail. A workable prototyping medium in this respect means that it is easy to leave out detail. This is true for paper but not for software prototypes, where rough aspects also require detailed specification; *all* elements require this.

This yields more work, and it explains the points that remain for relevance to account for: it makes building software prototypes harder to learn and to master, you have to learn and know how to produce these detailed specifications (knowing either programming or how to use a prototyping tool), which is obviously harder. It also diverts your attention from what you are designing to how you can get what you want (this is the 95% principle). Finally, because software prototypes require detailed specification of rough features, even these rough features are sensitive to small, technical details.

In contrast, using paper and pencil does not require detailed skills; you can focus on what you are doing rather than on how to do it; and paper prototypes are robust because no technical detail is involved. Hence, all of Rettig’s points can be traced to the relevance principle.

The twist with relevance is that it is directly linked to the degree of fidelity in a prototype. This should be somewhat obvious from the discussion about relevant detail. Between the two kinds of prototype of concern here, the main difference lies in their fidelity to the final product, and this is directly related to their relevance: A paper prototype looks much more unfinished than one built in software, which is more similar to the final product, and thus has a greater amount of the properties the final product will have.

This is just what the terms hi-fi and lo-fi refer to: a large amount of the final properties yields high fidelity with respect to the final design, and low fidelity from fewer details yields high relevance. (It is safe to assume that it indeed is the irrelevant properties that designers leave out.) Hence, it is the very fidelity itself that lies behind relevance. Thus, it is also the very fact that paper prototypes are lo-fi that makes them superior to software prototypes. This desire for low fidelity makes clear that these prototypes do not have a productive, but an inquiring purpose.

Goodness as an inquiring material

Even though building a prototype is also a cognitive activity, the question of how it functions as an inquiring material is more fundamental—that is, how well it serves as a basis for inquiry and interactive cognition. This is the genuine purpose for which the user interface prototype is built.

The major function of user interface prototypes in design inquiry is for *user testing*, which will be given a separate analysis in its own section below. For the present purposes however, it is sufficient to say that user testing serves to test how the interface design stands up under realistic circumstances, by having a representative user work with the prototype "live" and on a realistic task. This applies equally to the use of both kinds of prototype. The greatest difference is that as a software prototype runs on a computer, the user can interact with it directly, whereas there will have to be a person who "runs" a paper prototype, to simulate the responses to the user's actions.

In the following, when there are references to the "user", this implies the context of user testing, or possibly also more informal uses. For example, someone may just be asked to look at it or try it out without a formal procedure, and then to give their general opinion.

User testing is part of the designer's general inquiry into the problem domain. It is a test that focuses not on the user, but on the prototype itself. The user is brought in to make the test of the prototype more realistic. In theoretical terms, she serves as support, as an additional aspect of the recreated future situation of use.

Furthermore, since this kind of test is performed when the design process is still very much in progress, it is a kind of formative evaluation. The test gives the designer insights about her design, and it is performed to guide further design. The ultimate purpose of both kinds of prototype is to make user testing successful; to give the designer good and relevant feedback from the test. This is their purpose; it is for example not to represent the final design as accurately as possible.

The following is what Rettig writes about the effect that software prototypes have on user testing:

> *Reviewers* [i.e. users] *tend to comment on "fit and finished" issues.* You are trying to get feedback on the big things ... With a slick software prototype, you are just as likely to hear criticisms about your choice of fonts, color combinations, and button sizes.

The slickness of a software prototype, in other words its very hi-fi-ness, causes users to comment on small details. At the early stage when these user interface prototypes are made, the designer's concern is rather with the essential elements of the design:

> The designer wants to know whether the fundamental ideas work as intended, but hi-fi prototypes bring out the wrong kind of comments. If the fundamental ideas don't work, the designer wants to know this soon, before more time and effort is invested in a bad solution. Smaller details are easily fixed later on; deeper problems are not.

So a hi-fi prototype directs the user's attention to the wrong issues, but what is more, Rettig claims that paper prototypes do exactly the opposite:

> In contrast, the hand-made appearance of a paper or acetate prototype forces users to think about content rather than appearance.

So here we have a clear-cut opposition between the two kinds, regarding the same property and the same function, where the weakness in one is the strength of the other. The finished-looking appearance is a weakness in software prototypes, whereas the rough unfinishedness of paper prototypes is a very big strength in them. However, this effect applies not only to users that handle the prototypes. Rettig also states that they have the same influence on *designers* working with them: they too come to worry about typefaces, colors and layout, and about making screens and menus look good too early:

> On the back side of the same coin, developers easily become obsessed with the prettiness-power of a good tool, and spend their hours choosing colors instead of coming up with new ideas.

The reason why designers want feedback on the big issues is that these are what they are working on at this point. They should be working on the fundamental questions, because effort spent on details will be wasted every time a piece of work is thrown away or replaced by a new version. But software prototypes divert designers' attention, too, onto details that are of no concern yet. Also user feedback on the details of a design is wasted as soon as the design is changed.

The two kinds of prototype share the same function, but they obviously do not fill this function equally well. The difference between the two kinds that Rettig describes can be construed as one of *focus*: One kind of prototype makes both designer and user focus their efforts on the proper issues, whereas the other kind pulls their attention in unwanted directions, so that both users and designer instead waste their time on the wrong issues. Hence, according to Rettig, focus is the problem with high-fidelity prototypes in user testing, whereas the same issue is the strength of paper prototypes.

It is clear that Rettig's discussion is concerned with the quality of the feedback from user testing, the function of prototypes that I described above: he is explicitly discussing how the two kinds of prototype affect the feedback the designer gets, and this is also what his contrast between the two kinds is concerned with. Thereby, the advantages he is discussing are cognitive advantages. His points also concern the prototypes' cognitive impact on both designers and users, as inquiring materials that facilitate interactive cognition. Hence, what he is saying is that a good prototype is one that is good for cognition. There is reason to believe that something in the prototypes has a focusing effect on the cognitive performance of people using them; on their interactive cognition. More specifically, Rettig's descriptions indicate that focus is related to their degree of roughness and unfinished-lookingness.

Focus is made possible by having important aspects present in a working material, while leaving out those that are unimportant. A material that has only the right properties, and none other, will enable you to focus on the important ones. The rest are a potential source of distraction.

From Rettig's comments, it seems that users' attention is drawn to the most eye-catching features of a prototype such as the choice of colors. If these features are present, it is harder to focus on less conspicuous matters. In this case, the latter are the deeper, more conceptual questions that designers want users to concentrate on.

This is again what I previously described as relevance. Irrelevant properties serve no purpose in an inquiring material, but reduce focus; they should not be in a good prototype. In particular, it is again the level of relevant detail that plays a prominent role: A hand-drawn button with a scribbled label is only specified to a low level of detail. If it its drawn by a computer, then it is given a specific size and position, and the text is given a typeface, text size and perhaps a color. These are all added details, and they are just the things that Rettig claims have a bad influence on the attention of both designers and users. In the case of the designer's focus, this is also linked to the issues above, with the 95% rule in particular. Having to care about unnecessary details is a good example of how the designer's focus can be disturbed, whether it is by the workings of the tool or the specifics of the prototype.

Prototypes to experiment and explore with

Software prototypes are not only used for user interfaces. In software engineering, there is also a use of "pure" software prototypes. They have no physical embodiment but share the purpose of conventional prototypes, as programs that have a part of the functionality of the final product.

What makes software engineering prototypes interesting here, even though they are not entirely representative of prototypes in general, is the painstaking detail with which they have been documented. What other design discipline would devote entire conferences and books to prototyping alone (Bischofsberger & Pomberger 1992, Budde *et al.* 1992, Floyd 1984), not to mention the number of scientific papers on the topic?

This literature has made a distinction between prototypes that are used for exploration and experimentation (Bischofsberger & Pomberger 1992, Budde *et al.* 1992, Floyd 1984). Their meanings correspond to the sense in which I have used these two terms. I will draw upon some of this effort to give some further illustrations of the ways in which prototypes are used in design. Exploratory prototyping has been described in the following way:

> The goal of exploratory prototyping is to obtain a requirements definition that is as complete as possible and that can be verified by the later user on the basis of realistic examples. Its purpose is to permit the developers an insight into the application area, to allow them to discuss various approaches to a solution, and to clarify the feasibility of the proposed system in a given organizational environment.
>
> Beginning with initial conceptions of the proposed system, a prototype (of at least the user interface) is developed that makes it possible to test these conceptions on the basis of concrete examples and to successively (re)define the desired functionality. The important factors are not the quality of the prototype implementation, but the functionality, the ease of modification, and the speed of development ... Exploratory prototyping is an approach that supports requirements analysis and requirements definition. (Bischofsberger & Pomberger 1992, p 16–17)

I previously stated that a situating strategy is always performed within an inquiry, serving to make that inquiry possible. In this case, the inquiry concerns the analysis and definition of requirements; to develop the designers' understanding of what the authors refer to as "the application area" and "the proposed system in a given organizational environment". These are more abstract terms for what I have called "the future situation of use".

Exploratory prototyping is the activity that serves to make this inquiry possible: to "support" it, to "permit insight", to "allow discussion". These terms show the relation of dependence between the situating strategy and the inquiring function that it enables. The prototyping activity consists of exploring solution approaches and making a basic evaluation of them ("clarify the feasibility of").

Taken together, the pattern of exploratory prototyping is typical of situating strategies and how they are used: It is the prototype that makes these activities possible, as a material that enables interactive cognition. It "makes it possible to test these conceptions on the basis of concrete examples"; in other words, through working interactively and hands-on—or with software prototypes, at least hands-on-the-keyboard. The "basis of concrete examples" is named as important for yielding insight. Concrete examples stand in contrast to abstract specification, which is the norm in software engineering. This implies that such intramental, abstract reasoning is inferior to the interactive cognition that is made possible by the prototype.

The authors have here also embedded a comment on relevance as the principle behind a good prototype. They declare "the quality of the prototype implementation" to be unimportant. This quality refers to how good the program is according to the standards of software engineering, the standards by which "real" software, such as the final product, is measured. Roughly, this means how well the software conforms to the requirements specification. This includes how many bugs there are, how fast the program is, and so forth, but also how complete the program is, with respect to the specification. There is reason to believe that this is a major part of what the authors mean by "quality". For example they subsequently quote two other textbooks which state that a prototype is "not necessarily representative of a complete system" (Bischofsberger & Pomberger 1992, p. 19, Boar 1983, Connel & Shafer 1989). In any case, they state that all of these factors are unimportant. But it is also a remark that software prototypes have a purpose that is quite different from those programs that are the regular products of software engineering.

Rather than quality, the authors state that "the functionality, the ease of modification, and the speed of development" are factors that indeed are important. The two last points speak for themselves, development and modification are precisely the two aspects that were discussed earlier. "Functionality", finally, is what you can use the prototype for, the inquiring activity of exploratory prototyping that they are describing.

But the most conclusive part is how the relation between these aspects is stated. Important are not completeness and quality, which characterize the final product, *but* the three "important factors", precisely those factors that are crucial to a situating strategy: its function in inquiry, plus being easy to build and modify. This is precisely the relevance principle, where fidelity is traded for usefulness in inquiry, translated into the language of software engineering. There is a corresponding description of experimental prototyping:

> The goal of experimental prototyping is to achieve a concise specification of the components which form the system architec-

> ture. Its purpose is to experimentally validate the suitability of system component specifications, architecture models, and ideas for solutions for individual system components. ... [A] prototype is developed to permit the simulations of the interaction of the designed system components. ... Experimental prototyping is an approach that supports system and component design. (Bischofsberger & Pomberger 1992, p. 17)

Although this activity is somewhat different from the previous one, the description displays the pattern of a situating strategy: a software prototype of the system is developed; the purpose of the prototype is "to permit" simulations of the system and its internal workings. These simulations in turn provide the basis for experiments by which the designer can test the design. This experimental evaluation of the system is the inquiry, whose purpose is to work out the specifics of the design itself; the authors lay stress on the internal parts of the software system. This inquiry would not be possible without the situating strategy, which by means of simulations of a software prototype re-creates the future system. As the last sentence states, experimental prototyping is the basis for the design of the system.

The purpose of this procedure is appropriately described as experimental. It is explicitly described as being concerned with detailed tests—experiments—with the design itself and its internal workings. In contrast, the term "exploratory" is appropriate for the previously described procedure, since it was characterized as serving "to permit insight" and "allow discussion", rather than detailed and thorough testing. It is also exploratory in that it is used to understand the "organizational environment", not just "the proposed system"; the description of experimental prototyping is only concerned with the internal workings of the design itself.

Making prototypes function-relevant

So far I have only discussed the adaptation of inquiring materials in general terms, even though relevance is relative to purpose. This means that not all materials are made relevant in the same way. There may be two different prototypes of the same design whose differences make them better or worse for different kinds of inquiry. For this reason, adaptation can be taken further to match prototypes to their specific purpose. Thereby their relevance can be increased by matching their properties to how they will be used in inquiry.

Whereas the adaptation to purpose is specific to each individual case, the principle can be illustrated through how a prototype can be adapted to either exploratory or experimental use.

Relevance is achieved by leaving certain aspects of a design out of a prototype, and this can be done in two major types of ways: you can make either horizontal or vertical cuts, yielding *horizontal* or *vertical* relevance. The interest in the two kinds of relevance lies in the connection to their function and how they are used, because they correspond to the two activities of inquiry that have been discussed, exploration and experimentation.

Horizontal relevance is in fact what I introduced above to explain how you can gain relevance and focus on the higher level aspects of a design by leaving out detail. Although it is not immediately obvious, this is the kind of relevance that lies behind the roughness of a sketch or a paper prototype. Its crude-lookingness is a consequence of the minimal effort put into the sketch, not going beyond the most basic features. Hence, it is given only a minimum of detail, which is what horizontal relevance means.

Vertical relevance goes along the other dimension, working out certain aspects in full detail, while leaving others out entirely. These two kinds of relevance really correspond to two dimensions, both of which are involved in any actual prototype, as they are not exclusive in practice.

(This distinction derived from software engineering terminology, e.g. Bischofsberger & Pomberger 1992, Budde *et al.* 1992. They originate in the view of a system as a hierarchical tree of components: larger functionality higher up branches out into progressively finer detail downward in the tree, whereas related functionality is collected on the same branch. Vertical cuts remove entire functions by felling a whole branch, whereas horizontal cuts remove detail below them.)

The connection between the two kinds of relevance and use can be found in for example Tognazzini 1992. Horizontal relevance is best suited for exploratory purposes (*ibid.*, p. 81):

> Horizontal prototypes display most or all of the full range of the application ... without going in depth on any one part. Use to test the overall design concepts.

As the previous discussion showed, exploratory prototyping does not involve the specifics of a design. It serves insight, discussion and the development of ideas, rather than their detailed testing. It concerns "the overall design concepts", as well as the "organizational environment" that it will become a part of, and the fit between the two. Hence, effort spent on the details or internal workings of the prototype is wasted, it should be limited to the "outer" functional layers of the design.

As the purpose of experimentation is complementary to exploration, to test certain aspects in depth, a prototype that enables such testing should also extend along the vertical dimension (*ibid*.):

> In areas reflecting new design concepts and technology, build vertical prototypes that carry the user deep into the behaviors of specific parts of the system.

The discussion of experimental prototyping explicitly described it as involving detailed experiments with the design itself and its internal workings. Entirely new solutions are among the things that need to be tested in detail; for this purpose you need a vertical prototype.

The previous difference between thumbnails and roughs is also the same as that between these two kinds of prototype and their purposes. The small size of thumbnails serves to keep them undetailed and thereby horizontally relevant, and their use was previously characterized as exploratory. The connection between their form and function was also noted there, and this was really an early statement of horizontal relevance. Similarly, the purpose of roughs was identified as experimental, and also that their size is larger to accommodate a greater amount of detail; this is now recognized as the vertical relevance required for in-depth testing of a design.

Cognitive purpose of prototypes is poorly understood

Even with this array of evidence for prototypes as situating strategies—as materials created with a cognitive purpose in design—the perhaps best testimony is still how their productive properties are deemphasized in the literature:

> The important factors are not the quality of the prototype ... (Bischofsberger & Pomberger 1992, p. 17)

> Thumbnails are usually small because they are meant to be fast and undetailed. (Arntson 1993, p. 5)

> On the front [of a cardboard box] is written "desktop laser printer", that is all there is. It is a mock-up. The box is empty, *its functionality is zero*. Still, it works very well (Ehn & Kyng 1991, p. 171, my italics)

Sometimes you are even explicitly advised to avoid them:

> Construct a first version completely by hand. Sketch the widgets, hand-letter the labels. Don't even worry about using a straightedge at first. Just get the ideas down on paper. (Rettig 1994, p. 25)

This unfinished nature of prototypes, and the emphasis on keeping them unfinished, makes clear that they are not made to be the final product, since completeness and finality are explicitly given up. Thus, producing a prototype must serve the design process itself.

The same point is implicit in Brooks' (1975) famous advice to always plan to throw away the first version of a software program ("because you will anyhow"). If a program is created only to be thrown away, it purpose is obviously not to be the product of design. If it is still made, it must be it must be to the process of creation itself that it is important.

Nevertheless, as significant as this inquiring role is, it is still undervalued and poorly understood. In fact, only rarely is it even recognized for what it is. For example, software engineering texts regard prototyping mainly as a *technical* problem (e.g. Bischofsberger & Pomberger 1992, Budde *et al.* 1992, Floyd 1984). They are mainly concerned with how to produce prototypes; how to make them fit into established working procedures, or how these procedures should be modified to accommodate them; what tools are needed, etc.

I think the same oversight is at work when Rettig (1994) enumerates all the advantages of paper prototypes, and even so consistently refers to them as "lo-fi" prototypes. Low fidelity is all but a derogatory term, which measures them by how much they deviate from the final result. This is quite the wrong yardstick to measure them by. Instead, their name ought to reflect their strengths, because the issue is not how different they are from the end product.

Fidelity in prototypes comes from the amount of productive properties they are given, but what makes paper prototypes, thumbnails, and so forth so exceptionally useful is that these productive properties *are explicitly given up*. Instead of "low fidelity", something like "high relevance" or "highly useful" would be more appropriate, in recognition of their true cognitive purpose. Their present name merely proves the point that the cognitive purpose of paper prototypes is barely recognized.

Rettig, Ehn and Kyng, and others also note that paper prototyping is *fun*. I believe this is good evidence that it is an activity that lies close to what we and our minds are built to be good at. We enjoy informal, physical interaction with concrete materials, whereas abstract, symbolic reasoning is not quite so enjoyable. And "information processing" is the most boring work imaginable: processing forms, data entry, archival, record keeping, sorting, information retrieval, and so on and so forth. It is no coincidence, I am convinced, that we are "good at frisbee, bad at logic" (Edwin Hutchins, personal communication).

This is like how sugar, fat, and red meat (protein) are the foods that taste best. This is because they contain the most energy and were the best kinds of food in the era before supermarkets. Today, these foods are no longer those that are best for people, just as handling tools and holding things between two fingers are not the skills that employers look for anymore. Still, we should not forget that evolutionary pressures were different when our inherent capacities were once laid down, and that record-keeping skills did not impress a sabretooth tiger.

6.5 Scenarios

The use of prototypes by themselves does not diverge from the narrow view of design activity which considers the artifact only. However, prototypes are typically used together with other situating strategies, which cover other aspects of the future situation of use. As far as other physical materials are concerned, there is nothing that makes them different from prototypes besides not being the focus of design. When in the UTOPIA project an empty cardboard box was used as a mock-up of a laser printer (Ehn & Kyng 1991) it was not the printer that was being designed. Thereby the box was strictly not a prototype, although it worked like one, for all practical purposes.

But for other aspects, additional strategies are needed. Whereas a prototype embodies the focus of design, *scenarios* are used for recreating the wider situation around it. They are thereby closely linked to designers' concerns with more than the artifact alone; with its relationship to other parts of the future situation of use.

Dictionaries define a scenario as "the plot or outline of a dramatic work", or "a written version of a play, etc., in a film production, with

details of the scenes, etc." The use of scenarios in design has a great deal in common with scenarios in the original, theatrical sense. In a play, a scene is a single situation or sequence, and a scenario specifies the contents of such a scene. In design, the future situation of use is a scene where the artifact is being used, and a scenario serves to provide the details of such a scene, vividly enough to give a clear picture of what takes place there.

The primary benefit of scenarios is thereby *cognitive*, although this has hardly ever been spelled out explicitly. As with prototypes, the literature normally addresses practical, technical aspects. The exception is John Carroll's (1995) description, which captures the essence of scenarios remarkably well:

> The defining property of a scenario is that it projects a concrete description of activity that the user engages in when performing a specific task, a description sufficiently detailed *so that design implications can be inferred and reasoned about*. (p. 4, my italics)

And from the sentence that immediately follows, there can be no mistake that scenarios are a situating strategy (*ibid.*):

> Using scenarios in system development helps keep the future use of the envisioned system in view as the system is designed and implemented; it makes use concrete—which makes it easier to discuss use and to *design* use.

Carroll here makes the relation clear between how the scenario recreates "the future use of the envisioned system" and makes it *concrete*, and how this serves the designer's work: this future use becomes easier to discuss, to think about, and to reason about—and thereby easier to *design*. He also provides the following example:

- Harry, a curriculum designer, has just joined a project developing a multimedia information system for engineering education. He browses the project video history. Sets of clips are categorized under major iconically presented headings; under some of these are further menu-driven subcategories.
- He selects the Lewis icon from the designers, the Vision icon from the issues, and an early point on the project time-line. He then selects Play Clip and views a brief scene in which Lewis describes his vision of the project as enabling a new world of collaborative and experience-based education.

> ... [three more points similar to the second one, describing specific actions and their results] ...
>
> - Harry selects various combinations: other designers besides Walter and Lewis on these same issues and times, and other times and issues for Walter and Lewis. He begins to appreciate the project vision more broadly, attributing some of the differences to individual interests and roles in the project team. He begins to see the course topic issue as a process of discovering and refining new requirements through active consideration of alternatives. (Carroll 1995, p. 4)

On an immediate level, the scenario specifies exactly what is taking place in the scene itself: what people are present, what they do, in particular how they interact with the design itself, but also with other stage props (such as the cardboard box laser printer earlier), and of course, with each other. In this particular example, the script is simple in this respect; the only one present is Harry, and he is using the video database, performing the specified actions (selects Lewis icon, then "Play Clip", etc.).

But this is only one part of the scenario. The focus of interest is on how the design is used, and the implications thereof. However, in addition to the artifact, the user, and her actions, there are many additional ingredients that are essential in making the plot clear. Therefore, a scenario also provides a greater context beyond the scene itself, and which gives meaning to what happens there.

One dimension is the backgrounds of the people involved. This scenario details Harry's job title (a "curriculum designer", whatever that means), the project he is working on, and the design team he is part of. Another point of interest is education and relevant experience. This is only hinted at indirectly here, through Harry's job title.

The activity is also located in physical, organizational, social settings. Moreover, it is also part of other higher-level activities, here that of browsing the project history, which in turn is part of developing a multimedia system, and so on. Carroll's scenario also describes what the described events contribute to this bigger picture. Harry's individual actions are part of a browsing activity that serves to get him familiar with the team and the work done so far. This in turn gets him up to speed within the project. Other aspects that are specified may include a specific company, perhaps a certain department, and so forth.

There are also descriptions of the design itself, going beyond what a prototype can convey: "Sets of clips are categorized under major iconically presented headings...". Without these, the script would not make sense, with its details of what button is clicked when. So scenarios also provide meaningful context about the design itself. There are also other, technically irrelevant descriptions, such as those of the user's reactions and benefits (he reflects on the material, has insights, etc.)

This wider perspective makes sense of the specific events within the scenario, and these in turn give meaning to the design and its technical details. Even though these meaning-related aspects are immaterial, they would be available to the participants of a genuine use situation, just as much as would the concrete what's, where's and who's. Therefore, these are equally important parts of a naturally re-created future situation of use.

So scenarios may cover all dimensions of such a realistic setting: the people, the things, the events, and so forth. And for each dimension, they may include the concrete events as well as a wider perspective. This additional information is important in making the scenario meaningful, and to help the designer reason about it.

The all-embracing nature of scenarios has also been noted by Nardi (1992):

> An important feature of a scenario is that it depicts activities in a full context, describing the social settings, resources, and goals of users. It is not a narrowly focused task description but the "big picture" of how some particular kind of work gets done...

Concrete scenarios vs. abstract requirements

Mack also sees a major role as "the use of scenarios to represent the broader cognitive, social, and contextual aspects of work" (1995, p. 362), and that they thereby enable designers to bring these aspects to bear on design. He also notes that scenarios are suitable for "driving design activities". They thereby adopt the role that design methodology assigns to requirements specifications, as the vehicle of the design process. For example, scenarios are often used for the same purpose as such specifications, that is, to set the objectives that the design should meet. However, specifications and scenarios represent two diametrically opposed approaches to the same task.

The following is an excerpt from a prototypical requirements specification (from Guindon 1990b, p. 287):

4. All requests for lifts from floors must be serviced eventually, with all floors given equal priority (can this be proved or demonstrated?).
5. All requests for floors within lifts must be serviced eventually, with floors being serviced sequentially in the direction of travel (can this be proved or demonstrated?).

Here, as is typical of traditional specifications, the criteria are formulated with the purpose of covering all cases, and to be logically complete and exhaustive, and to do so they abstract away from any specific situation. However, they also give little or no clues about the specific operations in any particular case, or as to what kind of solution would meet these objectives. For instance, phrases such as "all requests must be given equal priority", or "all requests must be serviced eventually", do not say anything about how these criteria should be met, and the asking for a proof or demonstration shows that it is not even obvious whether a solution meets these criteria or not. There are also several abstract concepts, such as "eventually" and "equal priority", whose exact meaning must be determined before they can be used in design.

Therefore, to meet these abstract criteria, the designers in Guindon's study translated them into possible scenarios which they used instead. Guindon gives the following example of how one designer made up a simple scenario, and then used it to develop his solution:

> I'm not sure I understand about scheduling. I'll draw two elevators with a few floors. ... For each lift, I have, say, four buttons that are illuminated or not. And for each lift I also have to know the floor and the direction. Say Lift 1 is at floor 4 and there are requests to go down to floors 3 and 2. ... The floors don't move, the lifts move. It strikes me that I haven't considered enough this idea of having lifts between floors. I'm going to handle that. (Guindon 1990b, p. 286, also cf. figure 6.5)

Here, the designer translates the abstract requirements into specific conditions, by making up a simple scenario. He does this in order to *use* the requirements specification. (A closer analysis is given below.) But you also have to go the same way via concrete instances to *produce* it (this type of use, and others, are also documented in Carroll 1995). To ensure that the specification is complete and so on, the requirements will have to be compared against several specific test cases. One might wonder, then, why you should bother at all to go via the abstract formulation—in particular if the underlying scenarios aren't handed over alongside with it. Then the risk of translation problems seems apparent, not to mention the extra work required.

The desirable properties of a scenario

What makes designers prefer scenarios to traditional specifications, and what causes them to spontaneously create such scenarios if there are none? Having this side-by-side comparison with specifications in a specific case, it becomes easier to see precisely what it is that makes scenarios so cognitively useful: They are much like concrete examples of an abstract principle. Whereas requirements specifications are abstractions that require logical analysis, a scenario re-creates something that is as similar to the real-life situation as possible. It re-creates a future situation of use that is *specific*, *rich in details*, and *complete*, and thereby enables interactive cognition to work under the best possible circumstances. These attributes capture the essence of what makes scenarios cognitively useful.

Specific

The first advantage is that scenarios are *specific*. They always refer to a particular situation, with specific ingredients, whereas abstractions describe things in *general* terms:

> Scenarios should be as specific as possible, identifying by name real people and real companies that the team have in mind as prototypical users. (Tognazzini 1992, p. 74)

While abstractions refer to general classes, a scenario always has specific instances; Guindon's example turns "all requests for floors" into specific requests for the second and third floor, for example.

Going to specific instances

Going to specific instances is a well-known strategy in problem solving; Guindon provides an example of this strategy at work, when the designer is unsure of how to handle door control:

> In fact that insight suggests that the door control could be done

	Bob	Sarah	Earl & Stella	Dimitri & Melissa
Location	Los Angeles	Montana	Florida	Greece & Nevada
Age	35	52	70 & 62	24 & 22
Hobby	Work	Riding	Golfing	Hang gliding
Job	Investment banker	Horse ranch owner	Retired from insurance and teacher	Engineer and student
Car (in '92)	Mercedes	Range Rover	Lincoln Continental	Corvette (rent)
Income	High	High	Fixed	Overextended
Personality	A-type	Confident	Set in ways	Reckless
Gear	Communication equipment High tech	Dog, rifle	Toys for grandkids	Personal stereo
Misc.	Lives for work	Loves kids and horses	She teases re. his driving	On vacation

Table 6.1 Character map of a few imagined car customers.

> by a completely separate system from the handling of the service requests, but I'm not sure yet (Guindon 1990b, p. 286).

Instead of starting to work on a general solution from the beginning, he first tries to solve a specific case. He imagines how the doors would operate in a specific situation:

> Yes, in fact, usually in elevators first the doors open, then they stay open for a fixed amount of time, and then they close.

He thereby draws a specific conclusion about timing. Then he can use the insight to formulate a general requirement, and thus return to the initial, abstract problem of door control in general (p. 286):

> In fact, I should include a timer in the system controlling the opening and closing of the doors.

Compared to an abstract concept, a specific instance is easier to make sense of, to reason about, and to deal with in every respect. In this case we see how the designer conjectures that doors first open, then close. In the earlier example, the critical insight was: "The floors don't move, the lifts move." The value of trivial conclusions should not be underestimated.

Rich in detail

The second property of scenarios is that they are *rich in detail*. This also makes them more meaningful than abstractions. A situation that is rich in detail brings two kinds of advantages. For one thing, it makes the larger situation around the artifact available to design. But it also enhances the understanding of the design itself. This is because it allows you to study the design in its natural habitat, its interactions and relations with the surroundings, which brings out its natural patterns of behavior. Many of these will not be displayed when it is lifted out of its proper context, because there is nothing there which will evoke them.

There are other design strategies than scenarios that also serve to provide rich detail around the design. One such technique is *character maps* (table 6.1):

> One technique used by IDEO ... is the creation of *character maps*: *detailed* personality and activity descriptions for a small set of envisioned typical users. For example, in developing a product for automobile instrumentation, IDEO developed the characters of [table 6.1]. They are fictitious, created to cover *a broad range of the characteristics* that the team observed in the potential users of the product. Visualizing these characters helps designers to *anchor* their thinking about what their designs will mean *in practice* to the different people who may use them. (Winograd *et al*. 1996, p. 167, my italics)

The "broad range of characteristics" gives each character it richness in detail, and makes it possible to draw specific conclusions about how they will use the design, what they will want to do, problems that may occur, and so on.

Genuine

Creating specific, detailed circumstances such as these will also make the designer's expectations and ideas about use, problems of use, and so forth, more *genuine* and less contrived. It is a classical problem of usability design that designers are largely unable to place themselves in the shoes of the future users, and to imagine their genuine behavior to a sufficient degree:

> Because most designers have only limited contact with users … they simply do not realize how widely users differ, and, especially, how different many users are from most designers. … it is almost impossible to think about whether or not someone else will have trouble if you never encounter any yourself. In observing complete novices … we have often been amazed as they encounter major problems that we did not anticipate, or when problems that seemed simple to us were impossible for them to recover from. (Gould & Lewis 1985, p. 303)

Designers see themselves instead of actual users, and the behavior they expect from users rather than their genuine troubles. So their view of the future users and their behavior is prone to be contrived and unrealistic, unless they take explicit measures to avoid this. This is the purpose of creating specific and detailed descriptions of characters, use situations, and so on (Tognazzini 1992, p. 78):

> The key is to infuse the design team with vivid pictures of a series of prototypical users, so that the entire team will focus on designing for those people, not for themselves.

Working through detailed scenarios may for example allow a designer to discover things which would never have come across her mind otherwise: "The floors don't move, the lifts move. It strikes me that I haven't considered enough this idea of having lifts between floors. I'm going to handle that."

Complete

Designers' intuitions may therefore be described as *incomplete*. Rarely is only one scenario used, or one single character description, since each only covers a single case. Specific examples like these therefore usually come in groups, and thereby cover a range of specific instances. This will improve the *completeness* of the analysis, so that the eventual solution will be certain not to have left any blind spots (p. 74):

> Design with only a single user in mind, and you will find that only a single user can use your program. Scenarios force us to consider a wide range of users, in a wide variety of circumstances.

Scenarios approximate the physical world

So scenarios are specific, detailed, and so on. These are essential aspects of scenarios, but do not point at independent properties or separable dimensions. They all rather point to a complex of meanings blending together. For example, everyone knows what "concrete" means—at least we think we do, but if you ask yourself what "concrete" and "realistic" mean, and in particular what is the difference between them, then you probably see that their meanings are not very distinct. The same goes for "specific" and "detailed".

There are many other similar terms that capture the qualities of scenarios, but there is no selection of such terms without overlap. Instead I have tried to choose these terms to point out the most important aspects of this cluster of meanings. But what do they point at, what is it that they all have in common; that they all describe?

I suggest that all these properties can be traced back to the attributes of everything that is concrete and physically real, as opposed to abstractions and generic concepts. So scenarios are the closest possible thing to real life, and this is why they suit cognition so well. When the genuine situation is remote—not available or non-existent—the purpose of scenarios is to re-create it, in Technicolor, producing the conditions that interactive cognition is built for.

A fundamental thesis of interactive cognition is that mind complements, not replicates, the world. The world plays the role of itself. If the world's contribution is taken away, it becomes harder for the mind to do its share. A scenario serves to re-create the world's part of cognition, which complements the mind's contribution, and so allows the mind to work the way it is meant to. This is for example why Guindon's designer creates a concrete lift scenario, and draws it, to be able to think about the abstract requirements.

Hence, the properties I have enumerated all derive from co-presence and physical concreteness: In a real situation, every person is a particular one, and a real elevator is always in a specific location, not on floor n, where $1 \leq n \leq 4$. Also, a physical environment is always rich and detailed, unless an experimenter has removed everything that may yield context effects. Furthermore, it cannot be contrived, and nothing can be overlooked or forgotten, because there isn't anyone who must think of everything. Therefore a real situation is always complete, everything is there if only you look for it.

Vividness is a term that describes scenarios well. One might say that scenarios are so good because they are vivid; because they create such a strong impression of the vital issues, drawing out the distinct, tangible consequences of abstract requirements.

I would like to turn this argument around and suggest the converse: scenarios are vivid to us because they are good, "good for cognition" as it were. Since they match the abilities of cognition, they are able to deliver their message, and thereby create a vivid experience. Conversely, things that are not tuned to our frequencies fail to make a lasting impression. This does not merely entail loud noises and vibrant colors, but that the material complements the mind, and thus enables interactive cognition to work efficiently.

The principle of world complementing mind also means that specificity, richness in detail, and so forth, are properties that the mind does not need to supply, since the world usually does. There is plenty of support for this, the value of scenarios is one point in case, but also e.g. Reisberg (1987) has pointed out that we can easily form a mental image of a tiger, but we cannot count the stripes. Similarly, even though anyone can picture themselves a horse, drawing a horse's knee remains very difficult to most of us. Our mental images aren't specific and detailed enough, and so forth.

Norman (1993) has made a similar point about dreams. They don't follow the laws of physics; in dreams we can fly, walk through walls, people appear and disappear, and so forth. Norman argues that this is because these are constraints that the physical world imposes on us when we are awake, the mind doesn't have to do it.

There are also physiological facts supporting this. When we dream, the brain generates the same kind of activity as in the awake state. Importantly, it generates the same activity in the motor cortex as when we are awake and walk, and the perceptual areas react as when we actually see things, and so forth. What happens when we sleep is that the reticular activation system in the brain stem suppresses all in- and outgoing signals to the body (e.g. Luria 1973). What this means is that the physical constraints of nature disappear from the mental realm at the same time as the feedback from the physical world is cut off.

Norman also notes that the physical constraints are very resource-hungry parts of computer flight simulations. The point is not that the mind is like a flight simulator, but that enforcing the physical laws of nature is a very demanding task, and which the world usually does for free anyway.

Because of the way things are today, I feel obliged to point out that this is not a sign of the limitations of human cognitive capacity, for the mind should complement, not replicate, the world. If some researchers find that people are not good at some tasks, then it is because these tasks go against the nature of cognition, and therefore are improper measures of cognitive capacity. We should not accept them as limitations of human cognitive capacity, unless pigs' lack of wings also counts as evidence of what an inferior kind of bird pigs are; it is in other words a question of what yardstick youu use.

6.6 Simulation

Whereas a scenario usually has a narrative form and thereby includes the temporal dimension, it just specifies the events it includes. It merely provides the script, it must somehow be dramatized to come alive, to re-create the flow of time and events in a genuine situation.

The simplest form of doing this is by simulation, where the designer is re-creating—simulating—the future events by herself. In this sense, simulation is the simplest way of recreating a process, and less realistic than e.g. having real people actually perform the scene. For simple purposes, just reading the script to envision or imagine the scene may be sufficient. However, in design the situations that are re-created are usually complex enough to require paper and pencil, or more elaborate physical devices, in order to keep track of the particulars of the ongoing events.

Guindon has documented the use of simulation in the design of an elevator control program (Guindon 1990b, Guindon, Krasner & Curtis 1987). This was where the earlier example was taken from, which illustrated how a designer created a small scenario to make sense of the requirements for the elevator system. All the reported "simulations of scenarios" followed the same pattern. Their use was frequent, they were used by all participants at several points during design, and for several different purposes. In other words, they were important in all parts of the design process.

In the following example, the designer first sets up the scenario, and then simulates the events that follow from the initial conditions. He re-creates the flow of events by simulating one step at a time, considering the consequences of each one (Guindon 1990b, p. 287):

> I'm going to imagine one elevator and a few scenarios. Say there's a request from floor 2 to 4. If there is a lift going to 2 on its way up, then stop the lift at 2, open the doors, ... If there is a lift going down from 5 to 1, the lift does not stop at 2 ... What if you press up at the floor, but once in the lift, you press a down button. ... So there's definitively the need for a queue of lift requests for

> each lift, separate from the floor requests. … Maybe the floor requests could be handled by a completely separate system from the lift requests.

Could I borrow that pencil?

Guindon describes at length the designers' use of what she calls "external representations". Besides for such things as keeping notes and lists, their main use was for expressing the design-in-progress (*ibid.*, p. 290). But not merely for recording progress; the main thrust of Guindon's argument is in their role as the vehicle for simulations.

The simulations were made from an external point of view, incorporating the elevators, their positions and movements, and the button panels and displays both inside the elevators and by the elevator doors on the respective floors. All these aspects were included in the sketches that were used (cf. figure 6.5).

The information included in the drawings is external to the control program itself, and is thus not part of the design solution. These items would rather serve as external points of reference, as context in which the solution was grounded, and against which the developing was run and evaluated.

These drawings were used universally, and they were as central to the design process as the simulations they were used in:

> All three designers in this study supported the simulations of Lift scenarios by using external representations. (p. 287)

> All these simulations relied on external representations… (p. 291)

According to Guindon the designers' major reason for using "external representations" was "*difficulty in performing complex mental simulations*" (1990b, pp. 287–293, 1987, p. 75, also see *ibid.* pp. 69–70, 75–77). They were necessary even in the simplest simulations. This is also supported by the designers' comments:

> … it's kind of confusing, there's lifts (requests) and there's floors (requests) and it says "all requests for floors within lifts must be serviced eventually with floors being serviced sequentially (in the direction of travel)". Apparently that means … Let me give a better example … *I'll have to draw a picture.*

> Let's say the third guy wants to go the fifth floor. Let's say there's a floor request on the. Oh, I missed something here. Floor request has originating floor and direction… *Could I borrow that pencil?* (1987, p. 76, my italics)

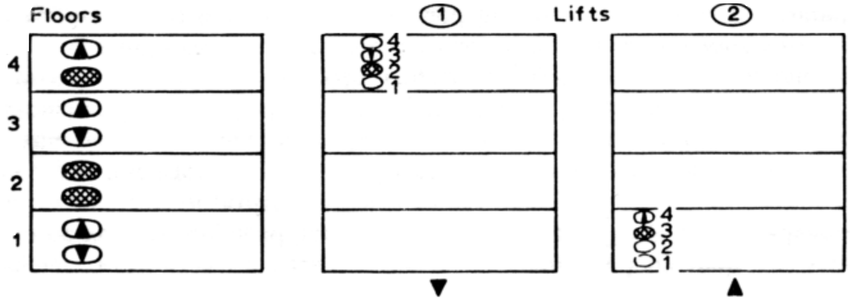


Figure 6.5 "I'm not sure I understand about scheduling. *I'll draw two elevators with a few floors.* … For each lift, I have, say, four buttons that are illuminated or not. And for each lift I also have to know the floor and the direction. Say Lift 1 is at floor 4 and there are requests to go down to floors 3 and 2. " (Guindon 1990b, p. 286, my italics)

The drawings would serve to anchor the simulation, and keep the designer's train of thought on the tracks. And though these sketches often worked, as they were not perfect for the task, not all problems with simulations were alleviated. For example, they remained shallow (1990b, pp. 291–293).

How mental are "mental simulations"?

It is therefore somewhat puzzling that Guindon, after stressing the necessity of using sketches, still persists in speaking of "mental" simulations. For example:

> By exploratory design, we mean design with many mental simulations of the problem environment and mental simulations of tentative solutions unguided by a plan. (1987, p. 68)

With this consistent use of "external representations", how "mental" can these simulations really be? The conflicting terminology is striking, verging on self-contradiction:

> Our designers had multiple uses for the external representations of their design solutions. … [One] important use was to support

> mental simulations of the solution, which would otherwise be too taxing cognitively and lead to breakdowns. (1990b, p. 290)

Further evidence for the non-intramental nature of such simulations can be found in Newell & Simon (1972). The three problem solving tasks that they studied—cryptarithmetic, the Moore–Anderson task, and chess—take up the majority of the 900-odd pages. However, there is one place where they describe genuine, non-laboratory problem solving. This happens in an addendum which is their own account of the early history of cognitive science. There, if only in passing, they describe their own work on developing the Logic Theorist (cf. chapter 1). In their own words, "on December 15, 1955, the first successful *hand simulation* was carried out" (Newell & Simon 1972, p. 883, my italics). This is in other words the same activity that Guindon described, and they obviously used the same technique as her subjects did. They also briefly mention that they missed a special test case when they performed the simulations manually, which was only discovered several months later when they first ran them successfully on a computer.

So the creation and use of drawings as material was essential for the successful use of these simulations. Nevertheless, Guindon persists in calling them "mental" simulations. How come?

I don't think that we should take her words too literally. "Cognitive" has historically been synonymous with "mental"—something is cognitive and therefore in the head, the reasoning would go. But there may be even less behind it, because traditionally no real distinction has been made between "cognitive" and "mental". There may thus be no real motive behind calling these simulations "mental".

And neither does Guindon argue that the simulations are strictly intramental—but evidently she gives a long array of evidence of the opposite. So she might just as well have called them *cognitive* simulations, and I believe that this is an appropriate description. The reason why I bring this up is that this fusion of the mental and the cognitive is very common, and I believe that little more lies behind it anywhere than that no one has asked themselves why or even if these terms are synonymous.

My point is that on closer consideration they are not synonymous; these cognitive simulations are better characterized as *interactive* than intramental. They adhere closely to the pattern of interactive cognition as presented in chapter 5: The sketches play the part of the world, the sketching activity is a form of inquiring action, and so on. This is the same point that Hutchins (1995, ch. 9) has made: the Turing machine (and subsequently the computer model of mind) is really not a model of the mind alone, but of Turing plus the pencil in his hand and the sketchpad on his desk, and the interactions among them.

"Mental" models

Moreover, I believe that "mental" as used in these cases is an epithet that originates from "mental models". Maybe cognitive science might have looked very different today had there only been a different word for "model"; if it had begun with an "I" or a "C" instead of an "M", for instance, then "mental" might not have been used so much.

As this connection implies, "mental models" are not particularly mental either. Observations of how people actually work do not support the idea of a mind that works well on its own. This time the support is easy to find, albeit in a somewhat surprising place. It has been there all the time, in the first contributing chapter of the classical book on mental models (Gentner & Stevens 1983):

> The models that people bring to bear on a task are not the precise, elegant models discussed so well in this book. Rather, they contain only partial descriptions of operations and huge areas of uncertainties. (Norman 1983, p. 8)

Norman also elaborated this argument in six points (*ibid.*), which I will discuss in greater detail. These points were presented as a list of problems with the then current theory of mental models; they were not compiled or presented as a coherent proposal or alternative. Rather, they were quite possibly regarded as assorted deficiencies of human cognition, which was a very common explanatory theme within the information processing theory of cognition. Only later would Norman propose an alternative explanation (Norman 1988), even though in retrospect, some of these six points foreshadowed what was to come.

For this reason, one would not expect these six listed problems to collectively point in one direction, or that they suggest or support a single, coherent alternative interpretation. Also, these points do not suggest such an alternative. Nevertheless, these six points can each by itself, as well as collectively, be interpreted as support for an interactive point of view:

- People's abilities to "run" their mental models are severely limited.

This point is general and rather vague, but since "running" a mental model must be very close to what Guindon calls mental simulation, it does echo the problems she describes, while also implying that they are related to a general cognitive principle: Our ability for intramental simulation is "severely limited", at least when we cannot employ physical materials as support.

The interactive explanation for this is that cognition is naturally adapted to work in interaction with the environment. If you remove the world's share in this, the result is the operation of half a cognitive system, which is also trying to compensate for this loss. For this reason, the mind doesn't behave like a computer, capable of imitating any process. The mind is built for operating *in* an environment, not for imitating or replicating it. Furthermore, by demonstrating that people can give decent "hand" simulations, given favorable conditions and suitable materials like paper and pencil, Guindon's study supports this interactive point of view.

Three of Norman's points are more specific, echoing the points made earlier with respect to scenarios:

- Mental models are incomplete.
- Mental models are unstable: people forget the details of the system they are using, especially when those details (or the whole system) have not been used for some period.
- Mental models do not have firm boundaries: similar devices and operations get confused with one another.

Both the first and second points state that the mind doesn't really store a copy of the world internally. The first one notes that mental models lack completeness, which reinforces the idea that completeness is a property of the world, and which the mind therefore doesn't need to provide. Above, completeness was identified as one of the major advantages of scenarios. Guindon also makes this point regarding simulations: Whereas designers had problems with maintaining the completeness of their simulations, Guindon lists this as one of the main issues that "external representations" provided help with, serving to "uncover missing information and to ensure completeness of the solution" (1990b, p. 290).

The second and third points state that the mind doesn't record detail, and that they therefore do not intramentally preserve the differences between similar objects. This too was listed earlier as a property of the physical world, which the mind therefore has no reason to duplicate. Norman has more recently made a similar argument, referring to a study by Nickerson & Adams where subjects were unable to distinguish genuine one-cent coins from ones where the details of their appearance had been altered (Nickerson & Adams 1979, Norman 1988). Norman's point was that to use these coins, we do not have to remember their exact layouts. We have to be able to tell one, five, and ten cent coins apart, but not to remember what the text says exactly, and where it is positioned. Consequently, he argued, our memories do not have to have great precision, and indeed problems arise when they must, such as with bank account numbers, PIN codes, social security numbers. In these cases, supplying precision—a responsibility that is normally assumed by the world—has been transferred to the mind. Then the capacities of an isolated mind are bound to appear limited. Hence, problems will also arise when objects that look similar are in fact different.

Norman's last two points bear most directly on the role of action:

- Mental models are parsimonious: Often people do extra physical operations rather than the mental planning that would allow them to avoid those actions; they are willing to trade-off extra physical action for reduced mental complexity. This is especially true where the extra actions allow one simplified rule to apply to a variety of devices, thus minimizing the chances for confusions.
- Mental models are "unscientific": people maintain "superstitious" behavior patterns even when they know they are unneeded because they cost little in physical effort and save mental effort.

These last remaining points apply more directly to the interactive perspective than the previous ones. Norman's paradigmatic example is that people tend to press the Clear button on calculators not once but even three times or more, always and regardless of whether it is necessary or not. The first point states the issue most clearly: Given that the functional unit is the mind alone, cognitive performance appears to come short: "extra physical operations" make up for insufficient mental planning.

Doubtlessly, in an intramental version, the cognitive task is to keep track of exactly which buttons to push, and then send these instructions to the fingers. But we should remember that the task at hand is not to plan what buttons to push, but to calculate the desired result. In an interactive version, the functional unit of cognition also includes action and world (button pressing and the calculator). Neither of the three parts in this functional unit is privileged or more correct, what matters is that the desired result is produced. Pressing Clear three or four times exploits the nature of calculators, to ensure that no previous calculations interfere with the present one: redundant pressing of Clear can have no unintended side-effects, whereas doing it only once definitely does, on calculators that require two key presses to erase all previous operations. It also exploits the nature of human action, in that tapping a finger rapidly several times is an automatic motor pattern, not more difficult than tapping it only once. So the mental part alone may appear imperfect, but the performance of the whole triad taken together is efficient, simple, and stable across varying conditions. Attaining the precision required—doing what you have to; being specific enough—is done not by the mind alone, but partly by action, and even partly by the calculator; each of the three contributing with one if its natural features that none of the others need to replicate.

Moreover, you gain a method that is fool-proof across all kinds of calculators—you don't even have to find out how the Clear button works if you don't know. In these cases the result is less work, by any measure: less key presses, shorter time, less mental effort, etc. This also includes the advantage from Tetris: physical action that is faster as well as easier than the intramental equivalent.

From this perspective, people are neither "unscientific" nor "superstitious". Their mental abilities are however calibrated for being a part of the whole interactive triad, and not the mind operating alone. Bearing these things in mind, letting the mental third of this remain "incomplete", "undetailed", and "without firm boundaries", makes *good sense*. It even makes the resulting whole somewhat more efficient. Therefore, the focus is on carrying things out interactively, and not in accordance to "the precise, elegant models discussed so well in this book". Still, it is somewhat odd that the arguments for the interactive nature of this modeling were provided in the classical text on mental models.

It is sometimes said that things outside the head cannot count as cognition, almost by definition and typically with regard to an approach like distributed cognition (e.g. Hutchins 1995). But consider what I have just said about "mental" models and "mental" simulations. These have always been regarded as cognition, and they refer to a certain classes of well-defined phenomena. But if now these phenomena were found to involve both action and world, would they then no longer count as cognition? Would it be said that only a third of "interactive" simulations is actually cognition? What is then the rest? This way of speaking and thinking about cognition takes some getting used to, but in the end I think it is our view of cognition that will change. I hope it will be defined in terms of function rather than location; as processes that gives us adaptive capabilities, rather than processes that are located within the brain.

Other situating strategies

Beside the situating strategies that I have described in this chapter, there are others which I haven't included here, but which still fit this category perfectly. Notable are those which serve to "re-create" the future *user* and *use activity*. Examples are user-testing and the "live" enactment of work situations that have been used in participatory design (as described by e.g. Ehn & Kyng 1991). In these, the future situation is reenacted by what are typically the actual users themselves. In order to set the stage for reenactment, these activities draw heavily on other situating strategies, for example prototypes and scenarios. They serve to make it easier for the "actors" to become as immersed in the conceived situation as possible, and to get them to act naturally, as they really would as users. The previously mentioned cardboard-box laser printer was a case in point, which made the participants actually walk to the printer to get their proof printouts.

There are relatively many, and quite diverse, situating strategies. The reason for this is that they together form an ecology of sorts, where the various kinds are suitable for re-creating different aspects of future situations of use. This applies for example to reenactment, as just discussed, but can also be seen for instance in Guindon's study, where drawings were used to represent the lift system (the artifacts), scenarios, to supply the specific conditions to test, and the so-called mental simulations, to recreate the process aspects of the future lift system in operation.

Hence, the diversity of the various techniques indeed serves a purpose, which is to fit the various niches that may need to be addressed, since also the varieties of future situations may be very dissimilar. An example of an unusual such situation was when Henry Dreyfuss (1955) and associates build a full-scale mockup of an airliner cabin, and re-created an entire eight-hour flight, including passengers with luggage, crews at work, and so on.

7. Intermission

7.1 Concerning the general validity of the theory

I have spent the preceding chapters making the case for design as being a cognitive process of a certain kind, but all this would be of limited interest if it applied only to design. Therefore, in this concluding chapter I will ask: How does my argument generalize? Does my exposition hold beyond the cases I have presented here?

It should first be noted, however, that the previous chapters have not been concerned only with design. On the contrary, it has been my aim to build a theory that is formulated on a general cognitive level, while using phenomena from design as the *basis* for this. Chapters 1 and 2 served to show the connection between design and the model of rational action, and thereby that the issues at stake in this book bear directly on the general views of cognition, in folk psychology as well as in scientific theories. Already in Dewey's original formulation, the theory of inquiry concerned cognition in general, and in chapter 3 the inquiring function of action and the "no pure analysis" conclusion were both formulated as general theories of cognition, as was the model of interactive cognition and the related concepts in chapters 4 and 5. The general relevance of chapter 6 will be discussed as part of what follows.

Thus, the preceding pages have not only been of relevance to design, even though one might at first think so. I will now also return to the bigger issues that I mentioned briefly in the Introduction, of whether cognition is intellectual or practical by nature. So how widely *does* my case for the practical view hold?

Let me briefly recapitulate the two contending views: In the view of cognition as basically *intellectual*, our most advanced cognitive capacities are the intramental ones; they comprise the highest achievement of evolution, and they are therefore what sets us apart from other species, and by which we do what no other animal can do. These are the capacities that have always been the main concern of cognitive science: mental representation, mental models, mental simulation; logical inference, case-based reasoning, planning, scripts, and so on.

In the other view, cognition is basically *practical*. There, also intellectual abilities are practical skills of a special kind, where extramental working materials and activities play key roles, and where these in turn are the products of cultural development and refinement. Hence, these abilities are not attributed to certain intramental equipment, as in the intellectual view. Surely the human brain goes beyond those of other animals, but to attribute our leg up only to mental modules that are uniquely human is too simple as an explanation, and too implausible as an evolutionary argument.

Defense #1: Intramental substitutes are used when interaction doesn't work

These are the two antagonists. The intramental model of cognition has always been advocated as superior, but this I have contested. Today it is old news that it doesn't work as advertised, but in addition to this I have sought to explain *why* it is so. And then I have presented a contrasting account of design as interactive and inquiring, but I have also made the case that because of its interactive nature, it is performed more effectively than it could be, even at best, if it followed the principles of traditional theory. These are two good reasons for cognition to be interactive rather than intramental. Thereby I have meant to, as it were, give cognition reason for being interactive.

But from this only, proponents of intramental theory will not accept that cognition is always interactive. The model I have presented gets its superior performance from drawing on working materials, action, and so forth, in the cognitive process. But this is not always possible; in particular, when the object of concern is not physically present and available to cognition. Let us call the favorable conditions *presence*, and the opposite conditions *remoteness*.

Being "representation-hungry"

A defense of the intramental view along these lines has been advanced by Clark & Toribo (1994) who cannot be considered to be conservative in this matter (pp. 418–419):

> The basic trouble ... is that the kinds of problem-domain invoked are just not sufficiently "representation-hungry". Instead they are, without exception, domains in which suitable ambient environmental stimuli exist and can be pressed into service in place of internal representations. ... it is unfair to use these cases to illustrate any more general anti-representationalist claim.

By a representation-hungry problem domain we mean a domain in which one or both of the following conditions apply:

1. The problem involves reasoning about absent, non-existent, or counterfactual states of affairs.
2. The problem requires the agent to be selectively sensitive to parameters whose ambient physical manifestations are complex and unruly (for example, open-endedly disjunctive).

Point one is a straight-forward definition of remoteness; Regarding the second point, the authors clarify that it mainly refers to cognition that deals with abstract properties that have no physically "manifest stimuli" (e.g. visual).

So, the argument goes, whereas cognition may well work interactively under presence, under remoteness it still relies on the full intramental capacities; indeed *must* rely on them, to compensate for these difficulties when remoteness prevents an interactive mode of operation. This is the most natural line of defense, and it is after all how intramentality and representation has always been motivated: It is superior because it can go beyond the here and now; beyond what is immediately present, beyond even what yet exists; to consider alternative, hypothetical situations, and so forth. This argument has been advanced several times, in modern day e.g. by Popper (1935/1959), who originally proposed that our hypotheses can die in our stead (also cf. chapter 5), Craik (1943), and Johnson–Laird (1983).

This *surrogate* capacity (cf. Clark & Grush in press) has a straightforward translation into the evolutionary claim: it is often held to be uniquely human or likewise; other species can purportedly also do the interactive things, but humans can do more, because they can use surrogates when the real thing is in limited supply.

Note, however, that Clark & Toribo have (on purpose) backed down from the fundamentally representational mind (also cf. Clark 1997). Previously, the advantages of mental surrogates have been taken as reasons for their being used by cognition *universally*, also when the original goods are readily available. However, with defense #1, a conservative position might even maintain that cognition still is intramental at the core, while conceding that it may take advantage of interaction when it can. The problem remains of explaining how cognition is sometimes interactive and sometimes intramental, but this is doable and merely a practical matter, even though as always with hybrid models, the result may not be aesthetically pleasing: Perhaps there are two systems that alternate, or maybe even the intramental processes are still running in the background during interactive work, maintaining an internal surrogate model of the environment even though it is not used just then.

By introducing the notion of "representation-hungry" domains, Clark & Toribo have established a clear line of defense, which is good since it identifies a meaningful direction in which to proceed with the argument. These hungry domains then ought to do the intramental capacities more justice, and there their utility should be evident, the authors claim: Whenever conditions degrade far enough, *then* cognition falls back on the representational, modeling, simulating system; when no real food for thought is available, then surrogates are provided for a mind that is starved.

Design is absolutely representation-starved

So arguments against intramentality should not concern interaction-friendly domains, i.e. presence, but representation-hungry ones —remoteness. Let us therefore be fair and do just that, because it can be done quite easily. For as I argued briefly in chapter 6, design *is* a representation-hungry domain; a closer look shows that the conditions for design are *just those* where representations are purportedly needed the most, where intramental cognition should work best, and therefore be of greatest value: Everything of that which is in the designer's chief concern is *twice remote*, in the future situation of use: It is *both* absent *spatially* and non-existent *temporally*. And perhaps it counts also as counterfactual, in that the whole point of design is to bring about something that today is not so. Clark & Toribo state that one of these conditions is enough to make design a hungry domain.

Design then qualifies as a perfect example of when mental surrogates should be of greatest utility, also in the classical view. The designer is concerned with a state of affairs that is distant in both time and space—and which Popper might well have called "hypothetical". And in many accounts, design is concerned with highly abstract properties, as in the second point above. Hence, it would be hard to think of a better match for their criteria on hunger. Design has also many times been explicitly stated as an important domain for intramental theory to explain in general; without reference to this particular issue (e.g. Akin 1986b, Newell & Simon 1972, p. 7, Simon 1973, 1981). It therefore cannot by any measure be regarded as an unfairly chosen domain.

Situating strategies: designers go out of their way to avoid intramental thinking

Hence, design is a perfect match of the conditions where the intramental capacities should come to their very best advantage, but in chapter 6 I showed at length what actually happens: Rather than falling back on intramental capacities, designers use *situating strategies*; they strive to restore or "re-create" presence, so that they can work interactively nevertheless. This goes to show that when hunger sets in, the switch from interactive to intramental cognition doesn't happen. And, after all, isn't it obvious that this is how it has to be? When you are starved, only *thinking* of food just makes you even more hungry.

There are two circumstances that make the force of this argument particularly strong and generally valid. Firstly, this is not a choice between equals, since the interactive alternative is not as readily available as the intramental procedure: To at all make it into an option, designers first have to use the situating strategies to *create* the working materials to interact with. Since they in this manner go *well out of their way* to enable the interactive mode, it means that the advantages of interactivity over intramentality are very strong (it might for example mean that the penalties of working intramentally are very high), which also increases the strength of this fact. So instead of falling back on intramental capacities, cognition spends *extra physical effort* with the only purpose of *avoiding* having to think intramentally.

The second "aggravating" circumstance makes a much stronger case for the *generality* of the interactive model: If cognition is not intramental even here, when the conditions for it are the best imaginable, then when would it be, since all other situations are *less* suitable than this one? Design is the purest possible manifestation of the exact problem that intramental theory was meant to solve; even in the words of the proponents themselves.

And in evolutionary terms, when these intramental abilities are in reason not used to perform the functions that are considered hard and uniquely human—the very functions for which they were once advanced—then when *are* they used? And conversely, when we rely on the older, more primitive interactive capabilities to perform the most difficult, hungry tasks, then what is the reason for having the advanced, intramental ones?

Hence, the "representation-hungriness" of design, and the use of the strategies to re-create presence, are strong cases in point for the *general* plausibility of the interactive model.

Defense #2: When also the strategies fail, then cognition becomes intramental

But this does not exclude quite all possibilities for intramental cognition. An ardent defender would still not yield, but might instead present a second counterargument: Maybe designers work interactively when they can, and when they can't do that, then they use the situating strategies; but when *they* break down, *then* the mental movie starts to roll. *This*, someone might claim, is when the intramental capacities come to their best advantage.

First, cognition would hardly be *fundamentally* intramental if this is the last way out; one might however still claim that intramentality, though not fundamental, provides us with capabilities for handling these extreme situations, which we otherwise wouldn't be able to deal with. But secondly, one then also comes to wonder what these circumstances would be: more extreme than those for which these intramental theories were created? In design for example, more extreme than design-by-drawing, so that the designer is forced to rely on abstraction and thinking only, without working on solutions, without involving users, and so on.

What comes to mind is a designer trying to comply with one of the systematic design methods from chapter 1. In terms of concrete working materials, and favorable conditions for interactive cognition, no approach to design could be more deprived than these methods: Design is inherently remote to begin with, and to a rather substantial extent, too. But on top of this, the approach that design methodology prescribes makes the inherently poor conditions even worse.

The objective of design methods is to focus on the abstract, logical structure of design problems, so from the view I have presented, their effect is essentially to deprive the design task to the fullest possible extent. From the ideal point of view, however, this approach is natural, with its emphasis on abstraction: The procedures that e.g. Alexander, Jones, and Simon advocated are concerned only with logical relations and abstract criteria like requirements and constraints—no drawings, nothing tangible, nothing concrete (Alexander 1963, Jones 1970, Simon 1981).

But from the interactive viewpoint, the goal that design methods aim for comprises the worst possible conditions for cognition, and the results have also been extremely poor; as seen in chapter 2, it is not only very hard to find successful examples, but even any examples at all, since the methods have proven so thoroughly impossible to use. Thereby the view I have advocated here would explain why design methods have yielded such disappointing results.

But to address this second defense on a general cognitive level requires a more elaborate discussion: it is a tricky matter, because the cognitive science tradition has made a diversive maneuver around the problem, rather than confronting it directly. And the diversion is large enough to require a section of its own.

Explaining (away) poor intramental performance

The evidence from design is quite conclusive that purely intramental performance is very poor, but the evidence is not restricted to design. On the contrary, this is probably the most well-documented fact in all of cognitive science: Innumerable experiments have documented so-called "cognitive limitations", "cognitive strain", etc. Whether the task studied has been concept formation (Bruner, Goodnow & Austin 1956), planning (Hayes-Roth & Hayes-Roth 1979), the comprehension of complex sentences, syllogistic reasoning (Johnson-Laird 1983), attention span, memorizing, mental models and mental simulation (cf. chapter 6 and Norman 1983)—the list could go on forever—whenever purely intramental performance has been studied, the result has always been that people do not perform according to the principles of intramentality.

Limitations on working memory: the tragical number seven, plus or minus two

But what is truly striking is the role this supposed limitation has come to take. It is not regarded as a measure for explaining (away) a theoretical mal-prediction, but as a celebrated scientific finding about the nature of human cognition—that is, the mind as having important limitations in its information processing capacity. This theme is so central that it has even been elevated into a general scientific principle, with its own element in the standard models of the human mind, known as *short-term memory* or *working memory*. Because of its role as the heart of all mental processing, every cognitive function needs to involve working memory in its operation. It is thereby the spider in the web of these models, which can easily be made into the fly in the ointment: If it is attributed with a flaw, then a problem in *any* cognitive function can be ascribed to this single defect.

This theme has been even further reified, by saying that working memory can hold only the legendary 7±2 items at a time. Equally legendary is the paper to which the origin of this fact is attributed, the full title of which is *The magical number seven, plus or minus two: Some limits on our capacity for processing information* (Miller 1956).

This is arguably the most well known, most popularized and most widely disseminated fact that cognitive science has ever produced, as can be seen in any popular scientific writing that touches on the subject of cognitive performance (Csikszentmihalyi 1990 is a prototypical case). Also, innumerable are the design features that have been claimed to "alleviate limitations in working memory"; citing this as the reason, for example, it has been stated that computer menus should hold no more than seven items for the user to choose between; it even seems that phone numbers were made sevenish digits long for this reason, so as to minimize the number of calls to directory assistance (Ellis & Beattie 1986).

Add to this the concept of a "chunk", which means that one of these seven slots can be said to contain anything, however large or small, complex or simple, and the size of which can be chosen so as to suit your purposes. Thereby you can maintain, for any given body of material, that it either fills up, overflows, or barely fits working memory—whichever you prefer (cf. Miller 1956, Simon 1974, 1976, 1979).

In this way the generally held principle, "limitations on human cognitive capacity", has been translated into its own architectural feature in these models, as a "limitation in working memory capacity", and has then been attributed to the specific size of this memory. It could arguably be maintained that the primary purpose of postulating this memory system has been to explain this "limitation".

Competence vs. performance

Closely related to this idea is another explanatory meta-theme which is also very wide-spread. This is the *competence vs. performance* distinction originating in Chomskian linguistics:

> A theory of the former would be a theory of linguistic knowledge and grammar, of what an *idealized* mature speaker-hearer of a language *could* say and understand; a theory of the latter would be a theory of behavior, of what *real* speaker-hearers *actually* do say and how utterances of others are understood. (Reber 1985, p. 137, emphasis added)

A better illustration of the contrast between the ideal and actual perspectives cannot be found. Chomskian theory portrays the human language faculty as based on exact rules that determine with great precision what language should look like. This theory is based on notions like formal grammars, automata theory, and other foundational theoretical principles of computer science. The problem is, of course, that this is not what you find if you look at the language that people use. To accommodate this circumstance, it is said that the theory concerns people's *competence*. This they are held to indeed have deep down below, but it isn't adequately revealed in people's behavior. That instead reflects their *performance*, which is much less sophisticated.

Now the crucial issue in this is to explain the *discrepancy* between competence and performance, and in particular how it can exist. This is done precisely by referring to people's inherent limitations in their "information processing capacity", and stating that the competence is degraded by the cognitive system that cannot fully handle it. This is often done with a direct reference to working memory capacity and the magic number.

Also the competence–performance theme has been used far beyond the original domain of linguistics, since it addresses a widespread and general scientific need. As a result, everything related to people's actual "performance" has attained a distinct derogatory ring to it. And since the study of performance is therefore not a study of cognition *an sich*, it has been regarded as "applied" science and of lesser value, since it tells us little about how cognition *really* is (and which we thus rarely see outside the laboratory).

Psychology as the study of human mental imperfection

Like so many other times, Newell & Simon have seen the weight and scope of also this matter clearer than many others, and they have addressed it on a general, domain-independent level, thereby making the issue very clear. But in so doing, they have also made the seriousness of the problem stand out more clearly, showing that it is not a minor issue with a restricted range of impact. In effect, their conclusion is that psychology is the study of human limitations and shortcomings:

> It is precisely when we begin to ask *why* the properly motivated subject does not behave in the manner predicted by the rational model that we recross the boundary again ... to a psychological theory of human rationality. The explanation must lie inside the subject: in limits of his ability to determine what the optimal behavior is, or to execute it if he can determine it. In simple concept attainment experiments, for example, the most important mechanism that prevents the subject from adopting an efficient strategy is usually the limit on the number of symbols he can retain and manipulate in short-term memory. To the extent that this is true, such experiments are experiments to reveal the structure of human short-term memory ...
>
> 1. To the extent that the behavior is precisely what is called for by the situation, it will give us information about the task environment. ...
> 2. To the extent that the behavior departs from perfect rationality we gain information about the psychology of the subject, about the nature of the internal mechanisms that are limiting his performance. (Newell & Simon 1972, pp. 55–56, referring to Bruner *et al.* 1956)

Here, they virtually *define* psychology as the study of how people fail to perform rationally; alternatively, of how they fail to behave as intramental theory says. Adequate performance is not a topic of psychology, it merely reflects the "task environment", but *poor* performance is: "when the subject does not behave as predicted by the rational model, we recross the boundary to a psychological theory". With such an assumption as a prominent part of the underpinnings, the scientific results are surely predisposed to point in a certain direction. There also seems to be a division of labor between different disciplines: one sets up the theories, the other documents how people fail to follow them. No room is then left for the theories to be influenced by how people actually behave; the ideal and the actual should not be mixed.

The same theme is discussed in Simon (1981), and it is also related to Simon's earlier work on "bounded rationality" and the concept of "satisficing", e.g. in Simon (1947). His distinction is probably based on the difference between e.g. economic models which often have assumed that the behavior of the involved agents is rational, and psychology as the study of how it isn't.

With respect to defense #2, the cognate themes discussed here all point in the same direction—"limited information-processing capacity"—from empirical results, to magic numbers, to competence vs. performance, all the way up to meta-scientific principles. The conclusion must then be that it is an *extremely* well-established fact that mental-only performance is quite circumscribed, not only in design, but in cognition in general: the mind alone does not reach any great heights.

Taken together, these "limitations" themes are how the ideal perspective accounts for the fact that cognition doesn't work as the theory dictates. Hence, this is a generally assumed position which appears even more ardent than the second line of defense I proposed above: It states that when the mind is forced to work intramentally, then it performs very poorly—still, it *is* in essence intramental. In this position, there is indeed a discrepancy between competence and performance.

7.2 A matter of choosing the proper yardstick

But now take a step back and look at what the "limited capacity" argument really says. First, cognition is held to work in a certain way; as a computer, intramentally, etc. Then, people are found not to behave as predicted, and quite thoroughly so at that. But the conclusion drawn is that they are information processors nevertheless—only *very bad ones*. The result is the following syllogism, which is the conclusion of defense #2:

The mind is a computer
It does not perform like a computer
∴ The mind is a *malfunctioning* computer

The logic behind this reasoning does comes across as slightly twisted: Is it the mind or the theory that should be sent in for repair? (Other variants are obtained by replacing *is a computer* with *is intramental* or *is rational*.)

Thereby, intramental theory faces two explanatory problems: In addition to the poor forced intramental performance, it should also account for how cognition can come to flourish in the presence of extra-mental factors. The sentiment is after all that the surrogate capacity should make the presence of external materials and so forth an insignificant matter:

> For the crucial activities ... take place centrally. This is true even when the desired object and the required activity are physical. (Newell & Simon 1972, p. 72, cf. chapter 4)

(But you already know the explanation: using the world alleviates the limitations on working memory.) Consider in comparison the explanation I have proposed. Concerning the latter question about the use of external material, I have already presented the interactive account at length. But what about the problematic performance on strictly intramental tasks? Quite simply, natural cognitive performance counts in three contributing parties: mind, world, and action. So what happens if you disable two of these? Of course, the performance breaks down. *But*, more importantly, this constitutes a highly contrived, unnatural task and thus very atypical performance: the tasks that are studied in the laboratory are very poor benchmarks to measure human cognitive capacity by. Therefore they are of limited interest. Psychological experiments do not eliminate the influence of situation and context, thus providing "generalizable" results, as experimental method holds; instead, also an experiment is a situation and context of its own, but of a very peculiar kind, and with very unrepresentative characteristics, yielding equally unrepresentative performance (cf. Whiteside & Wixon 1987, Wixon & Holtzblatt 1990). This can also explain why the experimental results have proven not to generalize to performance elsewhere.

As I see it, the main problem with intramental theory is its choice of *yardstick*: When the standard of measure is badly chosen, then the resulting measurements will necessarily be skewed. In the Introduction I gave a number of examples where human behavior had originally been framed as limited, substandard, and "irrational", but where others have later showed, by taking a different point of view, that human performance in these cases *makes sense*, even being clever and sophisticated. Compare with the examples given in the Introduction, e.g. the quotes regarding Micronesian navigation that were given there. My point is that the same is true here: When intramentality is the badly chosen norm, the resulting accounts of human performance will be biased by predestination; the resulting explanations will be in terms of deviations from the norm, that is, as limitations. Consider the following, admittedly somewhat drastic allusion:

> Our object of concern is human movement. The fundamental form of human movement is by flying. With the advent of modern aviation, we are able to give a detailed model of this capacity: we consider humans to be airplanes. Or more correctly, we consider humans, birds, and airplanes as three instances of flying systems.
>
> However, experimental studies have consistently shown that the human capacity for flying is limited—compared to for instance a Boeing 747–400. Much of this can probably be attributed to the human arms not being at all efficient as wings.
>
> The poor flying capacity is very well documented. For example, experiments off the Tower of Pisa, in the spirit of Galileo, have consistently shown that people can fly for only 7±2 seconds. Also, their landings are really messy. Accordingly, the study of human flying in effect becomes the study of human crashes.
>
> This also explains why we outside experimental settings mostly see people walking. Because of the limited abilities for flight, the legs that are really meant to serve during take-off and landing have become the major means of compensating for these limitations.
>
> This discipline has also, via knowledge of the limitations of human flying capacity, given us important guidelines that are most helpful for design. Some examples are: Build houses on or near the ground; use floors in rooms, particularly if the rooms are above ground level. Also place furniture on the floor, and door openings on the lower parts of walls.

Even though the points here may seem unjust, they have direct parallels in the cognitive literature: almost all of the statements in the first four paragraphs have direct counterparts in Newell & Simon (1972); some of them also e.g. in Miller (1956) and Card, Moran & Newell (1983). As the model for the last paragraph stands the advice found in *Applying Cognitive Psychology to User-Interface Design* (Gardiner & Christie 1987). The following are some examples of guidelines related to working memory:

- Working memory load increases the greater the amount of material that must be remembered temporarily, or "held in mind". (p. 159)
- If the number of referents, or the number of properties and relations ascribed to the referents used in a dialogue exceed the capacity available in working memory, then the probability of

considering all the relevant referents, or relevant properties and relations, will be reduced. (p. 117)

- If the sentences used in a dialogue are open to more than one interpretation, then they are read slower and less accurately than sentences that are open to just one interpretation. (p. 116)
- Do not present information that is *irrelevant* to the task that users are trying to perform. (p. 249)

In the truisms that are stated here, it seems that cognitive psychology is either completely absent as in the last two points, or else its technical terms have served to cloak the underlying self-evident truths with belabored sentences. Interestingly enough, the second point seems to say, "Complicated sentences are more difficult to understand." My personal favorites are however the following guidelines:

- hand-held devices, such as a mouse, light-pen or digitizing tablet-and-stylus combinations lend themselves ideally to drawing, pointing, selecting and moving tasks—in other words, spatial and visual tasks. (p. 271)
- keyboard-based commands are particularly appropriate for word-processing applications... Similarly, numerical data entry is best served by a keypad (*ibid.*)

But the original purpose of the flying analogy was to show how the chosen perspective can bias your vision, to the point where your view of things becomes outlandish. The whole theme of cognitive limitations was caused by the fatal choice of yardstick, which also lies behind the ill-conceived syllogism above: the yardstick is contained in the first line of the syllogism, which states that the mind is an information processor, and as such this chosen perspective functions as the axiom on which the conclusion will rest.

Also the *change* of yardsticks makes the difference between the two explanations: Given a more appropriate standard of measure, poor intramental performance is no anomaly, so the explanatory problem doesn't even occur. In the interactive view, intramentality is forced and unnatural, and with such restrictions the poor performance is to be expected. Hence, for one perspective the performance is coherent with the theory, but for the other it is an anomaly, and the difference results from the respective yardsticks. What makes the anomaly particularly serious is that it concerns not a minor or peripheral issue, but the main concern of the theory: intramental performance is anomalous to the theory of intramentality.

What the argument by Newell & Simon makes so clear is that the chosen yardstick can put you in a truly absurd situation, where theory forces certain decisions on you; here, having to define psychology as the study of how people fail to match the explanation you have chosen. As I see it, the main failure is that what is really a difficulty for your theory is made into a fault in the object of study. Consider physicists claiming that the deviations from their theoretical predictions were due to the universe committing "physical error", or limitations that give the world a restricted capacity to follow the laws of physics.

Probably the fatal standard of measure was adopted without much deliberation, or even seeing that alternatives existed. Soon, however, this created a situation where theory had to accommodate to the resulting measurements: concepts such as bounded rationality, limitations in cognitive capacity, the competence vs. performance distinction, working memory and its limited size, and so on. However, the premise in the above syllogism was never questioned.

But when we look back today, we can see that the premise is the cause of the problem. Because implicit in the view of the human as a rational, intramental computer was also the yardstick. And it was this yardstick that created the problem to explain: In reality, the question thought to be how people perform so poorly, was in fact the problem of why they perform so poorly *according to this yardstick*, which had been implicit in the view of cognition as rational and intramental.

Hence, it is worse not to realize that you are using a certain yardstick, than it is to deliberately have made a choice that turns out to have been bad. This is much like the matter of problem setting (from chapter 3): Instead of believing that your work is to produce the right answers to the only existing question, the most important insight is to know that what question you are asking will greatly influence the answers that result. The yardstick you are using, like a question and a problem definition, *sets* the types of result your work will produce. When you know that, the greatest part of choosing the right question or yardstick is already done.

7.3 From intramental functions to interactive technologies

I have here advocated a shift in perspective, from an intellectual/intramental view to an practical/interactive view of cognitive phenomena. If we are to make this shift, then what are the changes that will be necessary? How should the theories be changed? What will the results look like?

There already seems to be an emerging trend in which abilities, which were previously assumed to be intramental, are being reinterpreted as interactive and inquiring techniques. These developments are anything but concluded, and I can merely try to convey an idea of what may lie ahead of us.

Planning can serve as a good illustration. Ever since the birth of cognitive science, planning has been thought of as a fundamental cognitive function that is hardwired in the human information processor (Miller *et al.* 1960). According to Camhis (1979), the use of planning as a scientific concept began after World War II, in several different domains, urban planning as well as the philosophy of science, etc., compare with Polya (1945) and chapter 1. But since the emerging critique of cognitive planning theory, by in particular Suchman (1987), planning has begun to lose this status as a privileged and fundamental cognitive capacity. Instead, as for example in Agre & Chapman (1990), plans have come to take on the status of one cognitive technique among many, not more fundamental than, say, writing or riding a bike, but neither any less.

Spatial navigation and the use of maps are another example. For long now, animals as well as humans have been attributed with "cognitive maps" for navigational purposes. Gallistel (1990) represents this view, attributing them to animals as primitive as bees. Now there are indications that this attribution has been done too hastily. This was just what the Åkerlund quote in the Introduction concerned: since the Micronesian seafarers didn't navigate or use maps as we do, it was first concluded that they *couldn't* navigate at all. So just because bees find their way, they needn't use maps, because this is not the only way. For example, Hutchins (1983, 1995) showed that such things as viewer-independent perspective and certain representational techniques, which are required for creating maps, were invented in Renaissance times, are acquired by schooling, and do not even make sense to people in some cultures. Still, as was his point, their navigational feats are remarkable.

Hardware support for cognitive processing

Thus, there is an emerging pattern by which these sophisticated cognitive capacities, once thought to be naturally intramental and "hardware-supported" in the brain's computer, have begun to be reinterpreted as advanced cognitive *technologies*, highly developed working methods that are supported by likewise artifacts; *genuine* hardware support, that is. The procedures and artifacts alike have developed and matured over time, they in turn having been made possible by advances in mathematics and science. Hence, they didn't spring out of nothing, nor have they always been there, *in* there. Also, using them requires acquired, non-trivial skills.

This is what it means that also an intellectual ability is much a practical skill, but of a special variety. Plans are ways for organizing activities ahead of time, and they make use of technologies like writing and linguistic representations of activities, alphabets and so on, and inventions like paper and writing tools—all of which we take for granted. Taken together, these allow us to record, modify and reorganize resources for structuring activity, and later to use the resources to do just that.

The use of maps is even more obviously technological by nature. It involves the same writing tools, but also intellectual inventions like the birds-eye perspective, graphical abstraction and representations of the physical environment, 2D-projections of space, and also such recent advances as geometry, trigonometry, the Mercator projection and its likes, the whole domain of map-making methods and technology, the measurement and numerical representation of distances, and so on and so forth (see Hutchins 1995 for a comprehensive account). The use of maps comes natural to many people, particularly those with academic training, and even more so to those with a background in mathematics, geometry, and related disciplines like computer science—but not to everyone even in Western culture.

Design restored

What I have been attempting on these pages is, in a sense, a similar de-construction and re-construction of design, aiming to show that design is not a purely intramental process closely tied to the fundamental mechanisms of intentionality and planning, but a similarly sophisticated cognitive technology; developed over ages, and relying on subtle but sophisticated co-evolved artifacts and working techniques. Such is the combination of soft lead pencil, drawing paper, and techniques such as thumbnails, which together enable a highly fluid and expressive way of working that computers are far from matching.

A similar reinterpretation can be applied to for example mental simulation vs. hand simulation, as in chapter 6; we can do simulations reasonably well, given the proper supporting tools and techniques, but not so well otherwise. Some people have asked me, "Are you saying that we cannot plan?" No, I am saying, *But look at how we do it*; for example, look at how "mental" simulation is done. In comparison, am I saying that we cannot design, just because I argue that design is not a stage of pure intramental analysis, separated from the other activities of inquiry?

Hence, I don't claim that we cannot plan, but accounts of planning and so forth must be revised, like "mental" simulation. In one sense, design *is* the "restored" view of planning. "Plan" comes from the Middle French *plant*, which means ground plan or map (also influenced by Fr. *plan* as in flat surface, cf. English *plain*). In other words, from drawing a floor plan—i.e. architectural design. The term planning is thus an abstract rendition of design, derived from a process whereby you make plans literally by drawing. It appears that the practice of sketching as a means of design was developed at the same time as when design became a function separate from building (Gombrich 1960, Herbert 1993). Somewhere along the way, as "making plans" became intramentalized, both the working method and the materials and tools were dropped; they were in effect all made into epiphenomena—perhaps the tale of the singular creative idea was born here. Graphic designers also sometimes refer to sketching as "planning" the poster, folder, or other whatever they are producing (e.g. Black 1990). My account has thus merely reinstated the extramental components of the activity (inquiry), the doing and working techniques (sketching), and the materials and tools (paper, pencil), into the cognitive function of planning—such as it was in the original meaning of the word. One purpose of chapter 1 was to show how very closely related design is to general cognition, linked via the model of rationality and rational action.

In the same way, many other important activities have always been interactive and inquiring, but the intramental yardstick has caused this to be ignored. As compensation, mental mechanisms have been invented to handle what is actually done by interaction: representation and mental surrogates of the "outside" world are the paradigmatic examples (cf. Hutchins 1995, ch. 9).

It is such reconsideration that may lie ahead of us, for example regarding cognitive maps: the potential rediscovery of material and procedural factors that have been forgotten, and which would show that navigation, like planning and so forth, may not be as intramental as has been assumed. And if Micronesians navigate quite differently from Western seamen, then why do bees necessarily navigate with Western techniques?

But since these intramental mechanisms are part and parcel of cognitive science, the revisions will have to reach deeply into the groundwork. It will not be sufficient to for example talk about "external representations" and thereby keeping the theory of representation in place, with a new "external" specimen added to it. Hence, human abilities need to be explained differently, but also cognitive theory and explanation need to be reinvented. Instead of proposing new intramental innovations, we need to look for the answers elsewhere, in a very literal sense.

The final question

The topic in this last chapter has been whether cognition is intramental or interactive *in essence*; whether it is basically intellectual or basically practical. I have presented possible arguments and defenses from both camps, but there is one point which they agree on: As I have shown with the "cognitive limitations" theme, both sides agree that mental-only performance is underwhelming, and on this the evidence is decisive—*this* is not a matter of debate. So if you still wish to maintain that cognition is intramental, then you also need to adopt the "cognitive limitations" explanation. In interactive theory in contrast, the mind working on its own is only a circumscribed portion of the full cognitive system, and the unimpressive performance that has been documented so thoroughly is entirely to be expected.

Thus, between the alternative explanations of human performance on cerebral tasks, it doesn't come down to right or wrong, but to a matter of judgment. For example, the above syllogism seems to follow a rationality of its own: Would you say that this strange logic is due to limitations in the information processing capacity of its originators? I personally wouldn't. I'd grant them that they, too, would rather be playing frisbee.

Bibliography

ADELSON B & SOLOWAY E, 1985. The role of domain experience in software design. IEEE *Transactions on Software Engineering* 11(11), pp. 1351–1360.

ADELSON B & SOLOWAY E, 1988. A model of software design. In CHI M T, GLASER R & FARR M J (eds.) *The Nature of Expertise*, pp. 185–208. Lawrence Erlbaum, Hillsdale, NJ.

AGRE P & CHAPMAN D, 1987. Abstract reasoning as emergent from concrete acivity. In GEORGEFF M P & KLANSKY A L (eds.) *Reasoning about Actions and Plans: Proceedings from the 1986 workshop*, pp. 411–424. Morgan Kaufmann, Los Altos, CA.

AGRE P & CHAPMAN D, 1990. What are plans for? *Robotics and Autonomous Systems* 6, pp. 17–34.

ÅKERBLOM K, 1968. *Astronomy and Navigation in Polynesia and Micronesia*. Statens etnografiska museum, Stockholm, Sweden.

AKIN O, 1986. *The Psychology of Architectural Design*. Pion, London, UK.

ALEXANDER C, 1963. The determination of components for an Indian village. In JONES J C & THORNLEY D (eds.) *Conference on Design Methods*. Pergamon Press, Oxford.

ALEXANDER C, 1964. *Notes on the Synthesis of Form*. Harvard UP, Cambridge, MA.

ALEXANDER C, 1966. A city is not a tree. *Design* 206, pp. 44–55.

ALEXANDER C, 1971. The state of the art in design methods. DMG *Newsletter* 5(3), pp. 1–7.

ANDERSON J R, 1983. *The Architecture of Cognition*. Harvard UP, Cambridge, MA.

ARNTSON A E, 1993. *Graphic Design Basics.* Harcourt Brace Jovanovich, Orlando, FL.

ASIMOW M, 1962. *Introduction to Design*. Prentice Hall, Englewood Cliffs, NJ.

BATESON G, 1967. Cybernetic explanation. *American Behavioral Scientist* 10(8), pp. 29–32.

BISCHOFSBERGER W R & POMBERGER G, 1992. *Prototyping-Oriented Software Development: Concepts and Tools.* Springer-Verlag, Berlin.

BLACK A, 1990. Visible planning on paper and on screen: the impact of working medium on decision-making by novice graphic designers. *Behaviour and Information Technology* 9(4), pp. 283–296.

BOAR B H, 1983. *Application Prototyping—A requirements definition strategy for the 80s.* John Wiley & Sons, New York.

BØDKER S, GREENBAUM J & KYNG M, 1991. Setting the stage for design as action. In GREENBAUM J & KYNG M (eds.) *Design at Work: Cooperative Design of Computer Systems*, pp. 139–154. Lawrence Erlbaum, Hillsdale, NJ.

BOEHM B W, 1975. Software design and structuring. In HOROWITZ R (ed.) *Practical Strategies For Developing Large Software Systems.* Addison–Wesley, Reading, MA.

BRENNAN S E, 1990. *Seeking And Providing Evidence For Mutual Understanding*. Ph. D. dissertation, Stanford University, Palo Alto, CA.

BROADBENT G, 1973. *Design in Architecture.* John Wiley & Sons, New York.

BROADBENT G, 1979. The development of design methods. *Design Methods and Theories* 13(1), pp. 41–45.

BROOKS F P, 1975. *The Mythical Man-Month: Essays on Software Engineering.* Addison–Wesley, Reading, MA.

BROOKS R A, 1990. Elephants don't play chess. *Robotics and Autonomous Systems* 6, pp. 3–15.

BROOKS R A, 1991a. Intelligence without reason. In IJCAI*'91—International Joint Conference on Artificial Intelligence*, pp. 569–595

BROOKS R A, 1991b. Intelligence without representation. *Artificial Intelligence* 47, pp. 139–159.

BROWN J S & DUGUID P, 1994. Borderline issues: social and material aspects of design. *Human–Computer Interaction* 9(1), pp. 3–149.

BRUNER J S, GOODNOW J J & AUSTIN G A, 1956. *A Study of Thinking.* John Wiley & Sons, New York.

BUDDE R, KAUTZ K, KUHLENKAMP K & ZÜLLIGHOVEN H, 1992. *Prototyping: An Approach to Evolutionary System Development*. Springer-Verlag, Berlin.

CAMHIS M, 1979. *Planning Theory and Philosophy.* Tavistock Publications, London, UK.

CARD S, MORAN T P & NEWELL A, 1983. *The Psychology of Human–Computer Interaction*. Lawrence Erlbaum, Hillsdale, NJ.

CARROLL J M (ed.) 1995. *Scenario-based Design: Envisioning work and technology in system development*. John Wiley & Sons, New York.

CARROLL J M, 1997. Human–computer interaction: psychology as a science of design. *International Journal of Human–Computer Studies* 46, pp. 501–522.

CARROLL J M, THOMAS J C & MALHOTRA A, 1979. Clinical–experimental analysis of design problem solving. *Design Studies* 1, pp. 84–92.

CHI M T H, GLASER R & FARR M J, 1988. *The Nature of Expertise.* Lawrence Erlbaum, Hillsdale, NJ.

CHOMSKY N, 1957. *Syntactic Structures.* Mouton Gruyter, Paris.

CLANCEY W, 1997. *Situated Cognition: On human knowledge and computer representations.* Cambridge UP, Cambridge, MA.

CLARK A, 1997. *Being There: Putting brain, body, and world together again.* MIT Press, Cambridge, MA.

CLARK A & GRUSH R, in press. Towards a cognitive robotics. *Adaptive Behavior.*

CLARK A & TORIBO J, 1994. Doing without representing? *Synthese* 101, pp. 401–431.

CLARK H H, 1992. *Arenas of Language Use.* U. of Chicago Press, Chicago, IL.

CLARK H H, 1996. *Using Language.* Cambridge UP, Cambridge, MA.

CLARK H H & WILKES-GIBBS D, 1986. Referring as a collaborative process. *Cognition* 22, pp. 1–39.

CONNEL J L & SHAFER L B, 1989. *Structured Rapid Prototyping.* Yourdon Press, New York.

COOLEY M, 1988. From Brunelleschi to CAD–CAM. In THACKARA J (ed.) *Design After Modernism—Beyond the Object.* Thames and Hudson, New York.

CRAIK K, 1943. *The Nature of Explanation.* Cambridge UP, Cambridge, MA.

CROSS N (ed.) 1984. *Developments in Design Methodology*. John Wiley & Sons, Chichester, UK.

CSIKSZENTMIHALYI M, 1990. *Flow: the psychology of optimal experience.* Harper & Row, New York.

DASGUPTA S, 1989. The structure of design processes. *Advances in Computers* 28(1), pp. 1–67.

DENNETT D C, 1991. *Consciousness Explained.* Little, Brown & Co., Boston, MA.

DESCARTES R, 1628. *Regulae ad Directinem Ingenii (Rules for the Direction of the Mind).*

DESCARTES R, 1637. *Discours de la Méthode (Discourse on method).*

DEWEY J, 1903. *Studies in Logical Theory.* The U. of Chicago Press, Chicago, IL.

DEWEY J, 1925. *Experience and Nature.* Open Court Publishing Co., Chicago, IL.

DEWEY J, 1929. *The Quest for Certainty: a study of the relation of knowledge and action.* Minton Balch, New York.

DEWEY J, 1933. *How We Think.* Heath, Boston, MA.

DEWEY J, 1938. *Logic: the Theory of Inquiry.* H. Holt and Company, New York.

DEWEY J & BENTLEY A F, 1949. *Knowing and the Known.* Beacon Press, Boston, MA.

DIX A, FINLAY J, ABOWD G & BEALE R, 1993. *Human–Computer Interaction.* Prentice Hall, New York.

DONALD M, 1991. *Origins of the Modern Mind: Three stages in the evolution of culture and cognition.* Harvard UP, Cambridge, MA.

DREYFUS H L, 1991. *Being-in-the-world: A commentary on Heidegger's Being and Time, division I.* MIT Press, Cambridge, MA.

DREYFUSS H, 1955. *Designing For People.* Simon and Schuster, New York.

EASTMAN C M, 1970. On the analysis of intuitive design processes. In MOORE G T (ed.) *Emerging Methods In Environmental Design And Planning.* MIT Press, Cambridge, MA.

EHN P & KYNG M, 1991. Cardboard computers: Mocking-it-up or hands-on the future. In GREENBAUM J & KYNG M (eds.) *Design at Work: Cooperative Design of Computer Systems*, pp. 169–196. Lawrence Erlbaum, Hillsdale, NJ.

ELLIS A & BEATTIE G, 1986. *The Psychology of Language and Communication.* Weidenfeld & Nicolson, London.

ERICSSON K A & SMITH J, 1991. *Toward a General Theory of Expertise: Prospects and Limits.* Cambridge UP, Cambridge, MA.

FLOYD C (ed.) 1984. *Approaches to Prototyping.* Springer Verlag, Berlin.

FODOR J A, 1975. *The Language of Thought.* Crowell, New York.

GALLISTEL C R, 1990. *The Organization of Learning.* MIT Press, Cambridge, MA.

GARDINER M M & CHRISTIE B, 1987. *Applying Cognitive Psychology To User-Interface Design.* John Wiley & Sons, New York.

GARFINKEL H, 1967. *Studies in Ethnomethodology.* Prentice-Hall, Englewood Cliffs, NJ.

GENTNER D & STEVENS A L (eds.), 1983. *Mental Models.* Lawrence Erlbaum, Hillsdale, NJ.

GOEL V, 1995. *Sketches of Thought.* MIT Press, Cambridge, MA.

GOMBRICH E H, 1960. *Art and Illusion; A study in the psychology of pictorial representation.* Pantheon Books, New York.

GOODWIN C, 1979. The interactive construction of a sentence in natural conversation. In PSATHAS G (ed.) *Everyday Language: Studies in Ethnomethodology.* Irvington, New York.

GOULD J D & LEWIS C, 1985. Designing for usability: key principles and what designers think. *Communications of the* ACM 28(3), pp. 300–311.

GRAVES M, 1977. The necessity for drawing. *Architectural Design* 6, pp. 384–394.

GREENBAUM J & KYNG M (eds.), 1991a. *Design at Work: Cooperative Design of Computer Systems.* Lawrence Erlbaum, Hillsdale, NJ.

GREENBAUM J & KYNG M, 1991b. Introduction: Situated design. In GREENBAUM J & KYNG M (eds.) *Design at Work: Cooperative Design of Computer Systems*, pp. 1–24. Lawrence Erlbaum, Hillsdale, NJ.

GUINDON R, 1989. The process of knowledge discovery in system design. In SALVENDY G & SMITH M J (eds.) *Proceedings of the Third International Conference on Human–Computer Interaction*, pp. 727–734. Elsevier, Amsterdam.

GUINDON R, 1990a. Designing the design process: exploiting opportunistic thoughts. *Human–Computer Interaction* 5(2–3), pp. 305–344.

GUINDON R, 1990b. Knowledge exploited by experts during software system design. *International Journal of Man–Machine Studies* 33(3), pp. 279–304.

GUINDON R, 1992. Requirements and design of DesignVision, an object-oriented graphical interface to an intelligent software design assistant. In *Proceedings of* ACM CHI*'92 Conference on Human Factors in Computing Systems*,, pp. 499–506

GUINDON R & CURTIS B, 1988. Control of cognitive processes during software design: What tools are needed? In *Proceedings of* ACM CHI*'88 Conference on Human Factors in Computing Systems*, pp. 263–268.

GUINDON R, KRASNER H & CURTIS B, 1987. Breakdowns and processes during the early activities of software design by professionals. In OLSON G M, SHEPPARD S & SOLOWAY E (eds.) *Empirical Studies of Programmers: Second Workshop*, pp. 65–82. Ablex Publishing, Norwood, NJ.

HANSON K, 1969. Design from linked requirements in a housing problem. In BROADBENT G & WARD (eds.) *Design Methods in Architecture*. Lund Humphries, London.

HAYES-ROTH B & HAYES-ROTH F, 1979. A cognitive model of planning. *Cognitive Science* 3, pp. 275–310.

HEATH T, 1921. *A History of Greek Mathematics.* Clarendon Press, Oxford.

HEIDEGGER M, 1927/1962. *Being and Time.* SCM Press, London.

HERBERT D M, 1993. *Architectural Study Drawings.* Van Nostrand Reinhold, New York.

HESKETT J, 1980. *Industrial Design*. Thames and Hudson, London.

HINTIKKA J & REMES U, 1974. *The Method of Analysis: Its geometrical origin and its general significance.* D. Reidel Publishing Co., Dordrecht.

HIX D & HARTSON H R, 1993. *Developing User Interfaces: Ensuring Usability Through Product & Process.* John Wiley & Sons, New York.

HOLT J E, RADCLIFFE D F & SCHOORL D, 1985. Design or problem solving—a critical choice for the engineering profession. *Design Studies* 6, pp. 107–110.

HOLYOAK K J, 1991. Symbolic connectionism: toward third-generation theories of expertise. In ERICSSON K A & SMITH J (eds.) *Toward a General Theory of Expertise*, pp. 301–335. Cambridge UP, Cambridge, MA.

HULTSCH F (ed.) 1876–77. *Pappi Alexandrini Collectionis Quae Supersunt*. Vols. I–III. Weidmann, Berlin.

HUSSERL E, 1900/1970. *Logical Investigations.* Routledge and K. Paul; Humanities Press, London.

HUTCHINS E, 1980. *Culture and Inference: a Trobriand case study.* Harvard UP, Cambridge, MA.

HUTCHINS E, 1983. Understanding Micronesian navigation. In GENTNER D & STEVENS A L (eds.) *Mental Models*, pp. 191–225. Lawrence Erlbaum, Hillsdale, NJ.

HUTCHINS E, 1990. The technology of team navigation. In GALEGHER J, KRAUT R E & EGIDO C (eds.) *Intellectual Teamwork*, pp. 191–220. Lawrence Erlbaum, Hillsdale, NJ.

HUTCHINS E, 1995. *Cognition in the Wild*. MIT Press, Cambridge, MA.

JAMES W, 1907. *Pragmatism, a new name for some old ways of thinking.* Longmans, Green, New York.

JEFFERSON G, 1973. A case of precision timing in ordinary conversation: Overlapped tag-positioned address terms in closing sequences. *Semiotica* 9, pp. 47–96.

JEFFRIES R, TURNER A A, POLSON P G & ATWOOD M E, 1981. The processes involved in designing software. In ANDERSON J R (ed.) *Cognitive Skills and Their Acquisition*, pp. 255–283. Lawrence Erlbaum, Hillsdale, NJ.

JOHNSON-LAIRD P N, 1983. *Mental Models.* Cambridge UP, Cambridge, MA.

JOHNSON-LAIRD P N, 1989. Mental models. In POSNER M (ed.) *Foundations of Cognitive Science*. MIT Press, Cambridge, MA.

JONES J C, 1963. A method of systematic design. In JONES J C & THORNLEY D (eds.) *Conference on Design Methods*. Pergamon Press, Oxford.

JONES J C, 1970. *Design methods.* Van Nostrand Reinhold, New York.

JONES J C & THORNLEY D (eds.), 1963. *Conference on Design Methods.* Pergamon Press, Oxford.

KIRSH D & MAGLIO P, 1992. Some epistemic benefits of action: Tetris, a case study. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum, Hillsdale, NJ.

KIRSH D & MAGLIO P, 1994. On distinguishing epistemic from pragmatic action. *Cognitive Science*.

KRAUSS R M & WEINHEIMER S, 1964. Changes in reference phrases as a function of frequency of usage in social interaction: A preliminary study. *Psychonomic Science* 1, pp. 113–114.

KRAUSS R M & WEINHEIMER S, 1966. Concurrent feedback, confirmation, and the encoding of referents in verbal communication. *Social Psychology* 4, pp. 343–346.

LARKIN J, 1989. Display-based problem solving. In KLAHR D & KOTOVSKY K (eds.) *Complex Information Processing: The Impact of Herbert A. Simon*. Lawrence Erlbaum, Hillsdale, NJ.

LARKIN J H & SIMON H A, 1986. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science* 11, pp. 65–99.

LAVE J, 1988. *Cognition in Practice.* Cambridge UP, Cambridge, MA.

LAWSON B, 1980. *How Designers Think.* Butterworth Architecture, Oxford, UK.

LURIA A R, 1973. *The Working Brain—An introduction to neuropsychology.* Basic Books, London.

MACK R L, 1995. Scenarios as engines of design. In CARROLL J M (ed.) *Scenario-based Design: Envisioning work and technology in system development*, pp. 361–386. John Wiley & Sons, New York.

MALHOTRA A, THOMAS J C, CARROLL J M & MILLER L A, 1980. Cognitive processes in design. *International Journal of Man–Machine Studies* 12(2), pp. 119–140.

MARR D, 1982. *Vision*. Freeman, New York.

MILLER G A, 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review* 63, pp. 81–97.

MILLER G A, GALANTER E & PRIBRAM K H, 1960. *Plans and the Structure of Behavior.* Holt & Company, New York.

NARDI B, 1992. The use of scenarios in design. SIGCHI *Bulletin* 24(4), pp. 13–14.

NARDI B A & MILLER J R, 1991. Twinkling lights and nested loops: distributed problem solving and spreadsheet development. *International Journal of Man–Machine Studies* 34(2), pp. 161–184.

NARDI B A & ZARMER C L, 1993. Beyond models and metaphors: visual formalisms in user interface design. *Journal of Visual Languages and Computing* 4, pp. 5–33.

NEWELL A, 1990. *Unified Theories of Cognition*. Harvard UP, Cambridge, Mass.

NEWELL A, ROSENBLOOM P S & LAIRD J E, 1989. Symbolic architectures for cognition. In POSNER M (ed.) *Foundations of Cognitive Science*. MIT Press, Cambridge, MA.

NEWELL A & SIMON H A, 1972. *Human Problem Solving.* Prentice-Hall, Englewood Cliffs, NJ.

NICKERSON R S & ADAMS M J, 1979. Long-term memory for a common object. *Cognitive Psychology* 11, pp. 287–307.

NORMAN D A, 1983. Some observations on mental models. In GENTNER D & STEVENS A L (eds.) *Mental Models*, pp. 7–14. Lawrence Erlbaum, Hillsdale, NJ.

NORMAN D A, 1988. *The Design of Everyday Things.* Basic Books, New York.

NORMAN D A, 1993. *Things That Make Us Smart*. Addison–Wesley, Reading, MA.

PARNAS D L, 1985. Software aspects of strategic defense systems. *Communications of the* ACM 28(12), pp. 1326–1335.

PARNAS D L & CLEMENTS P C, 1986. A rational design process: how and why to fake it. IEEE *Transactions on Software Engineering* 12(2), pp. 251–257.

PEIRCE C S, HARTSHORNE C & WEISS P, 1931. *The Collected Papers of Charles Sanders Peirce.* Harvard UP, Cambridge, MA.

POLANYI M, 1958. *Personal Knowledge; Towards a post-critical philosophy.* U. of Chicago Press, Chicago, IL.

POLYA G, 1945. *How To Solve It*. Princeton UP, Princeton, NJ.

POPPER K R, 1935/1959. *The Logic of Scientific Discovery.* Basic Books, New York.

REBER A S, 1985. *The Penguin Dictionary of Psychology.* Penguin Books, London, UK.

REISBERG D, 1987. External representations and the advantages of externalizing one's thoughts. In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, pp. 281–293. Lawrence Erlbaum, Hillsdale, NJ.

RETTIG M, 1994. Prototyping for tiny fingers. *Communications of the* ACM 37(4), pp. 21–27.

RITTEL H W J, 1972. Second-generation design methods. *The* DMG *5th anniversary report:* DMG *occasional paper No. 1*, pp. 5–10.

RITTEL H W J & WEBER M M, 1973. Dilemmas in a general theory of planning. *Policy Sciences* 4, pp. 155–169.

ROBBINS E, 1994. *Why Architects Draw.* MIT Press, Cambridge, MA.

RUMELHART D E, SMOLENSKY P, MCCLELLAND J L & HINTON G E, 1986. Schemata and sequential thought processes in PDP models. In MCCLELLAND J L & RUMELHART D E (eds.) *Parallel Distributed Processing: Explorations in the microstructure of cognition, Vol. 2*, pp. 7–57. MIT Press, Cambridge, MA.

SACKS H & SCHEGLOFF E A, 1979. Two preferences in the organization of reference to persons in conversation and their interaction. In PSATHAS G (ed.) *Everyday language: Studies in ethnomethodology*, pp. 15–21. Irvington Publishers, New York.

SACKS H, SCHEGLOFF E A & JEFFERSON G, 1974. A simplest systematics for the organization of turn-taking for conversation. *Language* 50, pp. 696–735.

SCHEGLOFF E A, JEFFERSON G & SACKS H, 1977. The preference for self-correction in the organization of repair in conversation. *Language* 53(2), pp. 361–382.

SCHOBER M F & CLARK H H, 1989. Understanding by addressees and overhearers. *Cognitive Psychology* 21, pp. 211–232.

SCHÖN D, 1983. *The Reflective Practitioner.* MIT Press, Cambridge, MA.

SCHÖN D, 1987. *Educating the Reflective Practitioner.* Jossey-Bass, San Francisco, CA.

SCHÖN D, 1988. Designing: Rules, types and worlds. *Design Studies* 9, pp. 181–190.

SCHÖN D, 1992. Design as reflective conversation with the material. *Research in Engineering Design* 3, pp. 131–147.

SCHÖN D A & WIGGINS G, 1992. Kinds of seeing and their functions in designing. *Design Studies* 13(2), pp. 135–156.

SHRAGE M, 1996. Cultures of prototyping. In WINOGRAD T, BENNETT J, DE YOUNG L & HARTFIELD B (eds.) *Bringing Design to Software*, pp. 192–204. ACM Press, New York.

SIMON H A, 1947. *Administrative Behavior: A study of decision-making processes in administrative organization*. Macmillan, New York.

SIMON H A, 1973. The structure of ill-structured problems. *Artificial Intelligence* 4, pp. 181–201.

SIMON H A, 1974. How big is a chunk? *Science* 183, pp. 482–488.

SIMON H A, 1976. The information-storage system called "human memory". In ROSENZWEIG M R & BENNETT E L (eds.) *Neural Mechanisms of Learning and Motivation*, pp. 79–96. MIT Press, Cambridge, MA.

SIMON H A, 1979. *Models of Thought*. Yale UP, New Haven, CT.

SIMON H A, 1981. *The Sciences of the Artificial*. MIT Press, Cambridge, MA.

SPIEKERMANN E & GINGER E M, 1993. *Stop Stealing Sheep & find out how type works.* Adobe Press, Mountain View, CA.

SUCHMAN L, 1987. *Plans and Situated Actions: The problem of human–machine communication*. Cambridge UP, Cambridge, MA.

SWARTOUT W & BALZER R, 1982. On the inevitable intertwining of specification and implementation. *Communications of the* ACM 25, pp. 438–440.

TANNEN D, 1984. *Conversational Style: analyzing talk among friends.* Ablex Pub. Corp., Norwood, NJ.

TEGETHOFF W, DYCKES W & MIES VAN DER ROHE L, 1985. *Mies van der Rohe: the villas and country houses.* MIT Press, Cambridge, MA.

THOMAS J C & CARROLL J M, 1979. The psychological study of design. *Design Studies* 1(1), pp. 5–11.

TOGNAZZINI B, 1992. *Tog on Interface.* Addison-Wesley, Reading, MA.

VERA A & SIMON H A, 1993. Situated Action: A symbolic interpretation. *Cognitive Science* 17(1), pp. 7–48.

VON GLASERSFELD E, 1982. An interpretation of Piaget's constructivism. *Revue Internationale de Philosophie* 142, pp. 612–635.

WHITESIDE J & WIXON D, 1987. Improving Human–Computer Interaction: A quest for cognitive science. In CARROLL J M (ed.) *Interfacing Thought: Cognitive aspects of Human–Computer Interaction*, pp. 337–352. MIT Press, Cambridge, MA.

WINOGRAD T, BENNETT J, DE YOUNG L & HARTFIELD B (eds.), 1996. *Bringing Design to Software*. ACM Press, New York.

WIXON D & HOLTZBLATT K, 1990. Contextual design: An emergent view of system design transcending perspectives. In KNOX S (ed.) *Proceedings of* ACM CHI *'90 Conference on Human Factors in Computing Systems*, pp. 329–336.

ZHANG J, 1992. *Distributed Representation: The interaction between internal and external information*. Ph.D. Thesis, University of California, San Diego, San Diego, CA.

ZHANG J & NORMAN D, 1994. Representations in distributed cognitive tasks. *Cognitive Science* 18(1), pp. 87–122.