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.

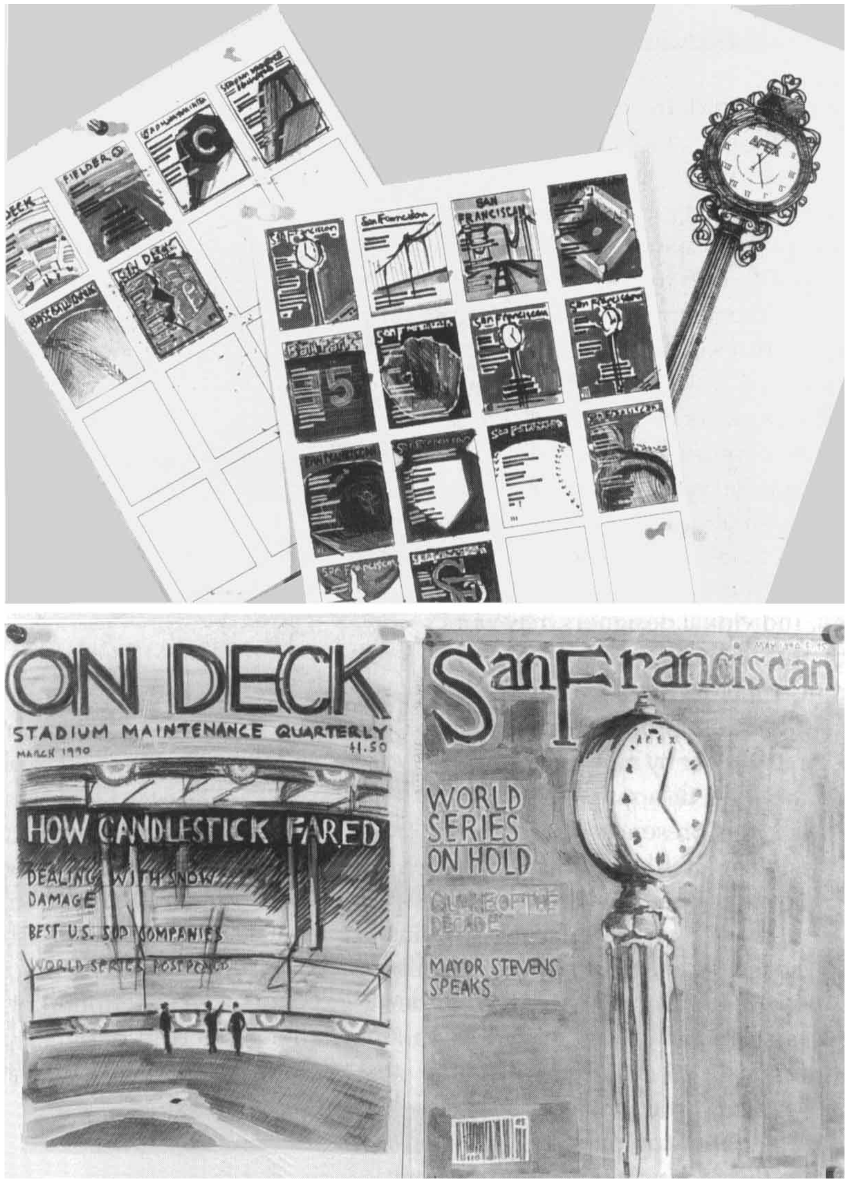

Figure 6.1 Two kinds of sketch for a magazine cover, with two different levels of precision and refinement. Note the relative sizes of each kind.

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)

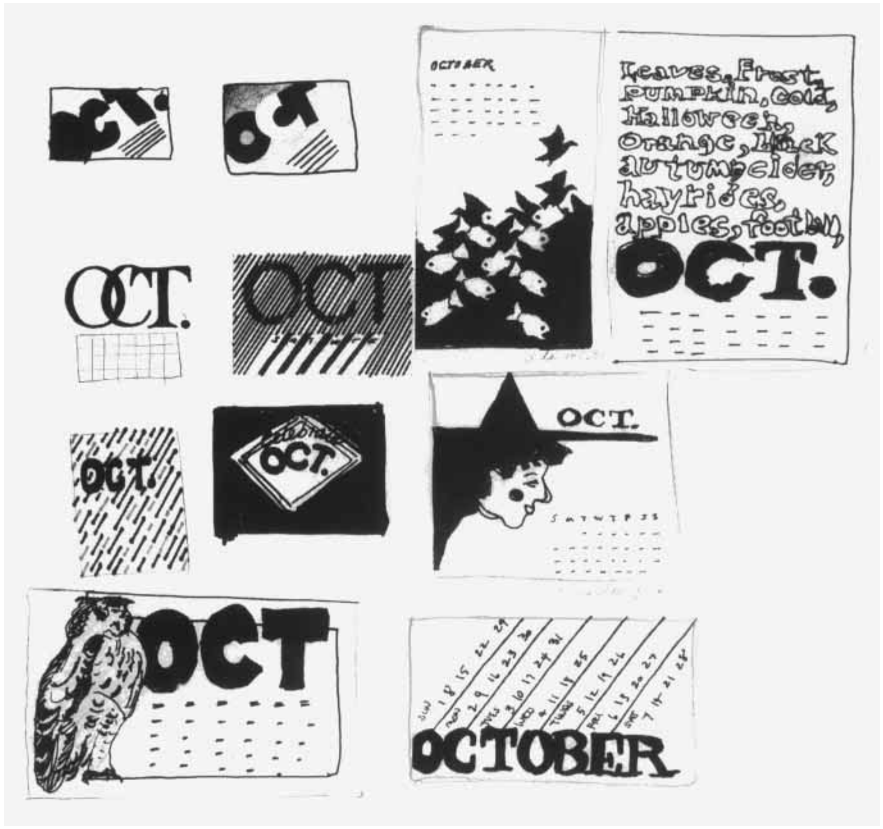
OCT.
OCT
OCTOBER
Leaves, Frost, pumpkin, Cold, Halloween, Orange, Black, autumn cider, hayrides, apples, football
OCT.
OCT.
OCT
OCT.
OCT.
OCT.
OCT
OCTOBER

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)

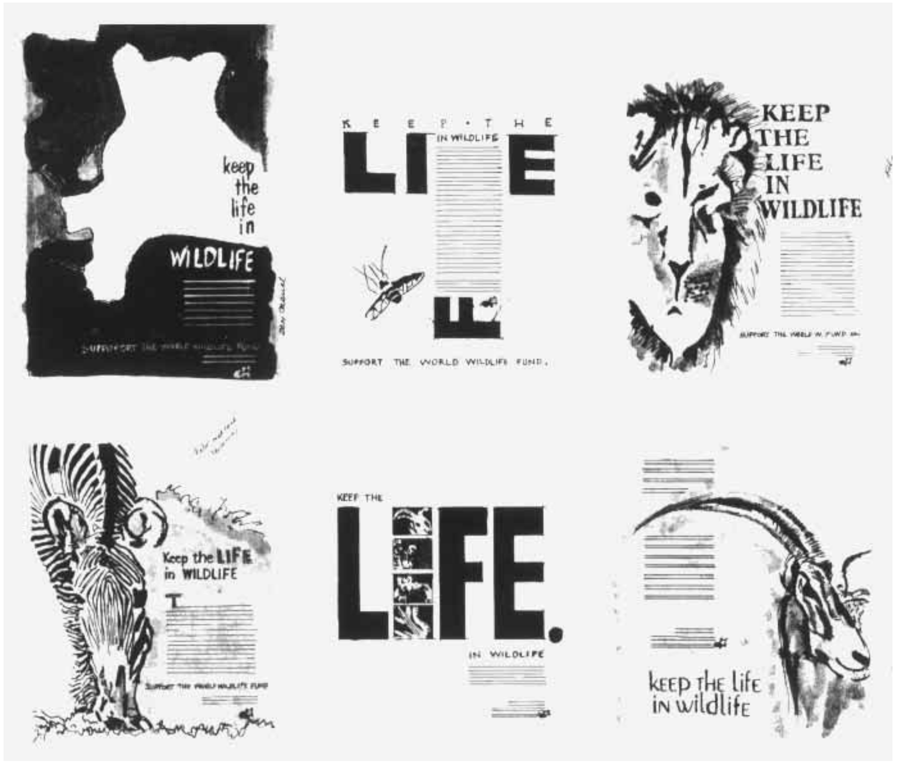
keep the life in WILDLIFE
SUPPORT THE WORLD WILDLIFE FUND
K E E P · T H E
LI E
IN WILDLIFE
SUPPORT THE WORLD WILDLIFE FUND.
KEEP THE LIFE IN WILDLIFE
SUPPORT THE WORLD WILDLIFE FUND
Keep the LIFE in WILDLIFE
SUPPORT THE WORLD WILDLIFE FUND
KEEP THE
LIFE.
IN WILDLIFE
keep the life in wildlife

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/non-present 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

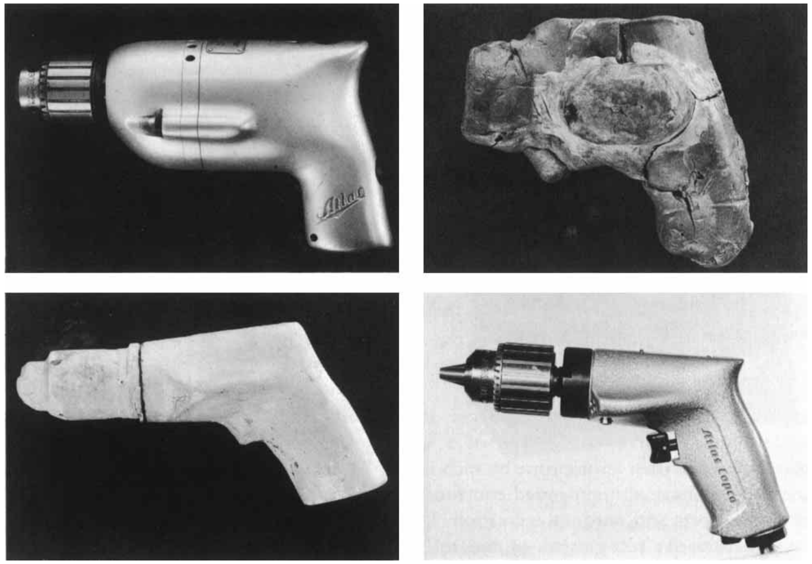

Figure 6.4 Industrial design prototypes and the final products.

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 sabre-tooth 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 re-creating 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 re-creates "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 $1 \le n \le 4$. 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

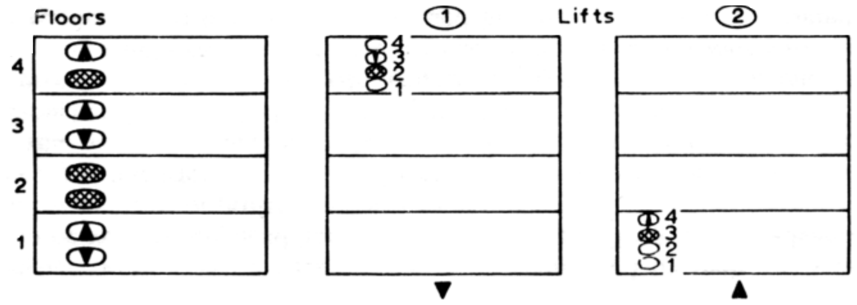

Figure 6.5 "I'm not sure I understand about scheduling. *I'll draw two elevators with a few floors.* ... For each lift, I have, say, four buttons that are illuminated or not. And for each lift I also have to know the floor and the direction. Say Lift 1 is at floor 4 and there are requests to go down to floors 3 and 2. " (Guindon 1990b, p. 286, my italics)

> 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)

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.