45. [Introduction] Cardboard Computers

Mocking-It-Up or Hands-On the Future

Some said Smalltalk went astray after Smalltalk-80, when it became much more difficult for twelve-year-olds to use the programming language. This complaint reflects the ideal that everyday users of new media should be able to design and create their own tools, specifically as embodied by the work of Alan Kay and Adele Goldberg (◊26), but more generally stretching back to the ability of users of Vannevar Bush's proposed memex to create their own collections of trails through information. Today this ideal is not much in evidence.

When ordinary users cannot create their own tools, of their own design, what options remain? Unfortunately, a common option is to embark on a quarterly-profit-driven tool creation process that leaves design as the last step, and stretch a pre-painted canvas over the underlying functions previously chosen by managers and implemented by programmers. Several essays in this book are dedicated to sketching out better alternatives: those by Ted Nelson (◊21), Nicholas Negroponte (◊23), Brenda Laurel (◊38), and Terry Winograd and Fernando Flores (◊37) in particular. All of these argue for beginning with design, and viewing design as more than a surface activity. Increasingly it is being accepted that design cannot be completed by a designer sitting alone. The design process must include users.

The standard way of involving users is in 'evaluation:' timing them at various tasks, asking them to react to various designs, and giving them surveys about their experiences to fill out.

Pelle Ehn and Morten Kyng have been at the forefront of a different approach. They are perhaps best known as leaders of the UTOPIA project (an acronym, in the Scandinavian languages, for 'Training, Technology, and Products from the Quality of Work Perspective'). UTOPIA is one of a number of projects that have taken the approach of working *with* users, from the outset, on the design of new media tools.

UTOPIA was carried out during a time of transition—in the early 1980s, before graphic user interfaces were widely available. Scandinavian newspapers were adopting computer page makeup tools developed in the U.S., which did not allow the graphic workers to use the full range of skills they had developed previously. As a result, the quality of Scandinavian newspaper design (which was different than U.S. design in certain ways) was diminishing, workers were losing their jobs in the wake of automation, and the quality of work life for those graphic workers who remained was deteriorating.

The unions decided to try taking the offensive. In Scandinavia there is a history of workers' groups becoming involved in research. Such groups had, for example, created the first ergonomic guidelines for computer display terminals. This time, the stakes were raised as the graphic workers union teamed up with university researchers to develop their own technological alternatives in the four year UTOPIA project.

The UTOPIA project's outlook was that new media tools should be designed for the quality of work they produce. As seen in this essay, Ehn and Kyng's model user is a skilled worker. The outlook is in some manners remarkably similar to Doug Engelbart's (◊07). Just as from Engelbart's perspective we would not expect the carpenter to use only tools that are 'user-friendly,' so Ehn and Kyng take the craft design process and craft use of tools as models of good software design.

Today, methods such as those used in UTOPIA are at the center of the Participatory Design (PD) movement, which has spread beyond Scandinavia to the U.S. and other parts of the world,

The 'tool perspective' is developed at more length in Ehn's *Work-Oriented Design of Computer Artifacts*. In a chapter titled "Tools," Ehn begins by defining tools in a manner distinguishable from other privileged human instruments (our bodies, our language, and our social institutions) in that they are designed, constructed, maintained, and redesigned. Next, Ehn enters a discussion of how computer tools should be designed—of what might make a good tool—from which some interesting parallels can be drawn out. For example, when Ehn writes, "A good tool becomes an extension of our bodies," (393) the obvious association from a new media perspective is to Marshall McLuhan's (◊13) concept of media as "extensions of man." McLuhan worked in a somewhat different portion of the political spectrum than does Ehn, however, which may lead one to wonder whether their politically-charged concepts are actually as similar as they sound. In this same discussion, Ehn's mention of good tools as "transparent," while inspired by philosopher Michael Polanyi, bears a similarity to the ideas of authors such a Donald Norman. It may therefore be open to similar critiques, such as the one mounted by Gregory Ulmer using the cinema studies concept of *apparatus* (see below). Later Ehn surveys the pitfalls and strengths of the tool perspective, and also takes up the implicit challenge offered by Brenda Laurel's (◊38) quite negative view of the tool perspective. Rather than argue that Laurel's view of computer use as mimesis is unhelpful, Ehn chooses to embed his view within her own, writing: "In a way a tool perspective is just a special case of designing computer artifacts as interactive plays—a case where the first-personness is carried out by a skilled tool user acting in a context of useful materials from which he or she can create good use quality products. It is a challenge to design, to create such tools for pleasurable engagement, tools that when used help the user transcend the boredom of machine work" (415).

07 13 21 26 37 38

. . . an experiment I conducted . . . placed the current developments in Artificial Intelligence and hypermedia programs in the context of the concept of the "apparatus," used in cinema studies to mount a critique of cinema as an institution, as a social "machine" that is as much ideological as it is technological. The same drive of realism that led in cinema to the "invisible style" of Hollywood narrative films, and to the occultation of the production process in favor of a consumption of the product as if it were "natural," is at work again in computing. Articles published in computer magazines declare that "the ultimate goal of computer technology is to make the computer disappear, that the technology should be so transparent, so invisible to the user, that for practical purposes the computer does not exist. In its perfect form, the computer and its application stand outside data content so that the user may be completely absorbed in the subject matter—it allows a person to interact with the computer just as if the computer were itself human" (*Macuser,* March 1989). It was clear that the efforts of critique to expose the oppressive effects of "the suture" in cinema (the effect binding the spectator to the illusion of a complete reality) had made no impression on the computer industry, whose professionals (including many academics) are in the process of designing "seamless" information environments for hypermedia applications. The "twin peaks" of American ideology—realism and individualism—are built into the computing machine (the computer as institution).

—Gregory Ulmer, "Grammatology Hypermedia"

and has had a profound influence on wider discussions of "usability." As this has taken place, similar methods have begun to appear in other contexts—some quite different from that of UTOPIA. For example, at SIGGRAPH 99 one designer presented his results from working with floor brokers at Goldman Sachs to design handheld computer tools modeled on their traditional skills, and operating as UTPOIA-style "reminders" of their work methods developed with paper tools. The brokers were quite enthusiastic about the results for their work life—showing it's not just union workers who would rather live in UTOPIA.

—NWF

From the Participatory Design Web page of CPSR who, along with organizations such as ACM SIGCHI and Xerox, have sponsored recent PD conferences:

Participatory Design (PD) is an approach to the assessment, design, and development of technological and organizational systems that places a premium on the active involvement of workplace practitioners (usually potential or current users of the system) in design and decision-making processes.

Because PD practitioners are so diverse in their perspectives, backgrounds, and areas of concern, there can be no single definition of PD. However, we can formulate a few tenets shared by most PD practitioners and advocates.

- Respect the users of technology, regardless of their status in the workplace, technical know-how, or access to their organization's purse strings. View every participant in a PD project as an expert in what they do, as a stakeholder whose voice needs to be heard.
- Recognize that workers are a prime source of innovation, that design ideas arise in collaboration with participants from diverse backgrounds, and that technology is but one option in addressing emergent problems.
- View a "system" as more than a collection of software encased in hardware boxes. In PD, we see systems as networks of people, practices, and technology embedded in particular organizational contexts.
- Understand the organization and the relevant work on its own terms, in its own settings. This is why PD practitioners prefer to spend time with users in their workplaces rather than "test" them in laboratories.
- Address problems that exist and arise in the workplace, articulated by or in collaboration with the affected parties, rather than attributed from the outside.
- Find concrete ways to improve the working lives of co-participants by, for example, reducing the tedium associated with work tasks; co-designing new opportunities for exercising creativity; increasing worker control over work content, measurement and reporting; and helping workers communicate and organize across hierarchical lines within the organization and with peers elsewhere.
- Be conscious of one's own role in PD processes; try to be a "reflective practitioner."

Further Reading

Bjerknes, Gro, Pelle Ehn, and Morten Kyng eds. *Computers and Democracy—A Scandinavian Challenge.* Aldershot, England: Avebury, 1987.

Computer Professionals for Social Responsibility (CPSR) <http://www.cpsr.org>

Ehn, Pelle. *Work-Oriented Design of Computer Artifacts.* Falköping, Sweden: Arbetslivscentrum and Hillsdale, NJ: Lawrence Erlbaum Associates, 1989.

Floyd, Christiane, Wolf-Michael Mehl, Fanny-Michaela Reisin, Gerhard Schmidt, and Gregor Wolf. "Out of Scandinavia: Alternative Approaches to Software Design and System Development," in *Human-Computer Interaction* 4(4): 253-350, 1989.

Paley, W. Bradford. "Handheld Interactions: Tailoring Interfaces for Single-Purpose Devices," in *SIGGRAPH 99 Sketches and Applications,* ACM Press, 1999.

Ulmer, Gregory. "Grammatology Hypermedia." *Postmodern Culture* 1(2). January, 1991.

Original Publication

Design at Work: Cooperative Design of Computer Systems. Ed. Joan Greenbaum and Morten Kyng. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1991. 169-195.

Cardboard Computers

Mocking-It-Up or Hands-On the Future

Pelle Ehn and Morten Kyng

Figure 45.1. Mocking-it-up.

This picture shows some artifacts we have used in designing the future of computer-supported newspaper production in the Utopia project. There are paper sheets on the wall, slide projectors, screens, racks, chip boards, some chairs, and a cardboard box. However, something is missing. No, it is not computers, but the empty chairs certainly have to be occupied by future users. In our view, artifacts, computers as well as other tools, should be understood via the human use of them.

The users who would be envisioning their future work situation in the design game above are typographers and journalists working with editing, layout, and page make up. The relationship between these two groups has always been a bit tense. Journalists (assistant editors who work with editing and layout of text and pictures are responsible for the quality of the content of the product (the readability); typographers (make up staff) who work with page make up are responsible for the quality of the form of the product (the legibility). However, the border between the two responsibilities is far from clear.

Figure 45.2. Page make up work using lead technology.

In the good old composing days journalists worked in the newsroom and typographers worked in the composing room. The editor sent a layout sketch to the composing room and the make up staff returned a "proof" to the newsroom before sending the page to be printed. Not a perfect process, but it worked well. With paper paste up technology it became more difficult to make proofs. It was too expensive and took too much time to run a proof on the photo typesetter. Hence, the assistant editors began to hang around in the composing room controlling the work of the make up staff. Not surprisingly, the typographers were not too happy about this arrangement.

With the introduction of computer-based layout and page make up in the late 1970s, the relations between the "rucksacks" (as the typographers called the journalists in the composing room) and the make up staff got even worse. Now the work was literally taken away from many typographers, since the equipment was placed in the newsroom and was operated by the assistant editors. However, aside from the personal misfortunes of introducing this new technology, the solution was far from optimal in terms of typographic quality.

The design question we were facing in the Utopia project was the following: Are there technical and organizational design alternatives that support peaceful and creative coexistence between typographers and journalists, where both readability and legibility of the product could be enhanced?

Now take a closer look at the cardboard box at the right in the first picture. On the front 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 in the design game of envisioning the future work of assistant editors and make up staff. It is a suggestion to the participating users that an inexpensive computer-based proof machine could be part of the solution. With the help of new technology, the old proof machine can be reinvented and enhanced.

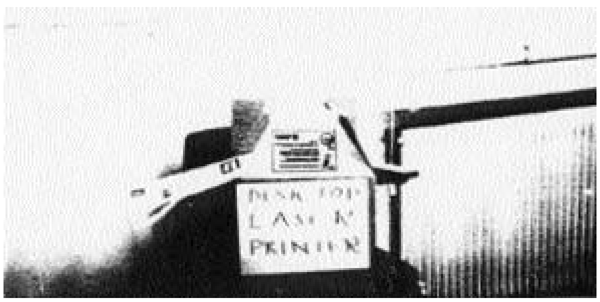

Figure 45.3. A mock-up of a laser printer “reinventing” the old proof machine.

The journalist makes a layout sketch, sends it to the typographer, and the typographer works on the page make up. Whenever he is in doubt or has suggestions for alternatives, he sends a proof via the desktop laser printer to the journalist, who marks with a few pen strokes how he wants the page to look, and sends the proof back to the typographer, who completes the page. Both can concentrate on what they are best at: the assistant editor on journalistic quality and the make up person on typographic quality. And why should they not sit in the same room and talk to each other?

The mock-ups in the pictures were made and used in 1982. At that time desktop laser printers only existed in the advanced research laboratories, and certainly typographers and journalists had never heard of them. To them the idea of a cheap laser printer was “unreal.” It was our responsibility as professional designers to be aware of such future possibilities and to suggest them to the users. It was also our role to suggest this technical and organizational solution in such a way that the users could experience and envision what it would mean in their practical work, before too much time, money, and development work were invested. Hence, the design game with the mock-up laser printer.

In this chapter we will show some prototypical examples of mock-ups. We will discuss how and why they are useful in participatory design. Our examples will range from “cardboard computers” to “computer mock-ups” hinting at the pros and cons of less and more advanced artifacts for envisionment of future use. Finally, the use of mock-ups is put into the perspective of other activities going on in participatory design.

Why Mock It Up?

What we suggest in this chapter is that design artifacts such as mock-ups can be most useful in early stages of the design process. They encourage active user involvement, unlike traditional specification documents. For better or worse, they actually help users and designers transcend the borders of reality and imagine the impossible.

But why do mock-ups work despite their low functionality and the fact that they only are a kind of *simulacrum?* Some of the obvious answers include:

- they encourage “*hands-on experience*,” hence user involvement beyond the detached reflection that traditional systems descriptions allow;
- they are *understandable,* hence there is no confusion between the simulation and the “real thing,” and everybody has the competence to modify them;
- they are *cheap,* hence many experiments can be conducted without big investments in equipment, commitment, time, and other resources; and last but not least,
- they are *fun* to work with.

We Did Not Make It Up

Certainly we did not invent the idea of using mock-ups. Kids have always been good at playing with mock-ups like dolls, cars, etc. It is hard to imagine human life without these kinds of games.

However, the use of mock-ups can be most seductive. Think of computer exhibitions. What looks like a running system is often not the final system, nor even a prototype, but simply a video tape or a programmed slide show. Good envisionment of a future product; however, more than one manufacturer has passed the border between concerned marketing envisionment and deliberate manipulation.

Our way of using mock-ups has a family resemblance to both children’s play and envisionment at exhibitions, but the most important inspiration comes from industrial designers. They have been using mock-ups professionally for decades. In particular, they have been successful in using mock-ups in ergonomic design.

Figure 45.4. Mock-up or the real system? An advertisement for the TIPS page make up system which was based on UTOPIA specifications. When the ad was published no "real" system existed.

One example is the use of ergonomic rigs. This is a mock-up environment in which designers and users together can build mock-ups of, for example, a future work station. Typically there will be support for rapid and cheap mocking-up of ergonomic aspects of appropriate tables, chairs, monitors, etc. Several alternatives can be designed and the users can get hands-on experience. Later, the designers can elaborate the mock-ups as in the following picture, where a future reception workstation has been envisioned.

Figure 45.5. Industrial design mock-up.

But it is not necessary to be a professional industrial designer to make useful mock-ups. The next picture is from a newsletter published by one of the clerical worker unions in Sweden. It shows a mock-up of a proposal for a new computer-controlled parcel sorting workstation. Originally, the local union was presented with only the technical specifications of the new proposal. However, on the basis of the drawings, the workers were unable to judge the quality of the proposal with respect to the effectiveness of work procedures and physical strain. They then spent a few thousand dollars to build the full scale mock-up. Using this they were able to simulate the future work: the flow of parcels, the tasks of each operator, including work load, and the possibility of supporting each other when bottlenecks occurred. The simulations resulted in several improvements, including suggestions for reducing physical strain and new ways of cooperating.

Figure 45.6. Sort machine mock-up. The headline reads: "We did not understand the blueprints, so we made our own mock-ups."

The use of mock-ups described here resembles the way industrial designers use them. However, our focus is on setting up design games for envisionment of the future work process. In contrast to industrial designers, we focus more on the hardware and software functionality of the future artifacts and less on the ergonomic aspects. Industrial designers often make very elaborate aesthetic and ergonomic designs of keyboards, but the display is black, and no functionality is simulated or mocked-up. If these different capabilities could meet in a participative design effort, an even more realistic simulacrum could be created. If the future users also actively participate in the design, the mock-ups may be truly useful and a proper move toward a changed reality. But are mock-ups really professional design artifacts? Yes, they are. In arguing this point, we will get a bit more philosophical, but we will also look at the theory in some practical examples.

Language Games

We are guided by the concept of "What a picture describes is determined by its use." This is shocking statement for those of us who were brought up in a natural science tradition where a system description normally is understood as a kind of mirror image of reality. Nevertheless, this is a position at the heart of Ludwig Wittgenstein's *Philosophical Investigations* (1953). Wittgenstein was aware of this challenge. As a philosopher, he was first known for writing a doctoral thesis that showed how with an exact language we can map reality (Wittgenstein, 1923). Then he spent the rest of his life trying to convince us that he was wrong—that there is more to human language and interaction than can be written down, and that language is action. Instead of focusing on mirror images of reality we are advised to think of the language games people play—how we are able to participate in human activities because we have learned to act according to the unwritten rules of that activity.

For example, if Pelle, a designer, points at the cardboard box with the sign "desktop laser printer," and says to Jon, a typographer, "Could you take a look at the proof coming out from the desktop laser printer," Jon does not answer "There is no desktop laser printer! You are pointing at an empty cardboard box, stupid." Rather he would go to the cardboard box, pick up a blank paper from a paper stack beside the box, turn toward Pelle, look at the paper, and say "Well, here we have a problem. There is too much text for a 48 point three column headline and that picture of the president. I think we will have to crop the picture, or the headline has to be rewritten."

According to Pelle, and the other participants, Jon makes a proper move in the design *language game* he is participating in. On the other hand, if Jon had maintained that there is only a cardboard box with a sign on it saying "desktop laser printer," he would have made an incorrect move in this specific design language game. Despite the fact that he would be right, he would not have understood how to play according to the rules.

The reason that Jon, Pelle, and the other participants can use the mock-up in a proper way is because this design language game has a *family resemblance* with other language games they know how to play. The language game in which the cardboard box is used has a family resemblance with the use of a traditional proof machine in the professional typographical language game which Jon knew very well, as well as with technical discussions Pelle had participated in as part of his profession. Furthermore, they both know how to play this design language game using the mock-up, because it resembles other *games* they have played before.

However, cardboard boxes do not become laser printers by themselves. In fact, one of the hardest challenges for the designer seems to be to *create a design language game that makes sense to all participants;* the designer in the role of play-maker. In this role the designer sets the stage by finding and supporting ways for useful cooperation between professional designers and "designing users." Future workshops and metaphorical design, as well as organizational design games, are examples of ways to set the stage for such shared design language games. Mock-ups and prototypes may be useful "properties" in these games. Hence, mock-ups are only effective in the design language games that make sense to the participants. In these games mock-ups play an important role as something to which one may refer in discussions of the design; as a *reminder* pointing back to experience from using the mock-up. Thus, instead of having to produce rational arguments in support of a certain point of view concerning a breakdown in the use of a mock-up, it is possible to repeat the sequence of operations leading to the breakdown. Then both that situation and the steps producing it may be evaluated, alternatives tried out, and, if necessary, participants may try to give rational arguments in favor of their point of view.

In summary, mock-ups become useful when they make sense to the participants in a specific design language game, not because they mirror "real things," but because of the interaction and reflection they support (see Ehn, 1989).

A new role for the designer is to set the stage and make it possible for designers and users to develop and use a common situated design language game. This has to be a language game that has a family resemblance with the ordinary language games of both the users and the designers;

a language game which is socially constructed by the participants.

Hands-On Experiences and Ready-to-Hand Use

There are, however, more to mock-ups and the language games in which they are used than just language. As opposed to linguistic artifacts, such as flowcharts and system description documents, mock-ups make it possible for the user to get *hands-on experience.* This is illustrated in the picture below. What you see is the first mock-up we ever made in a design language game.

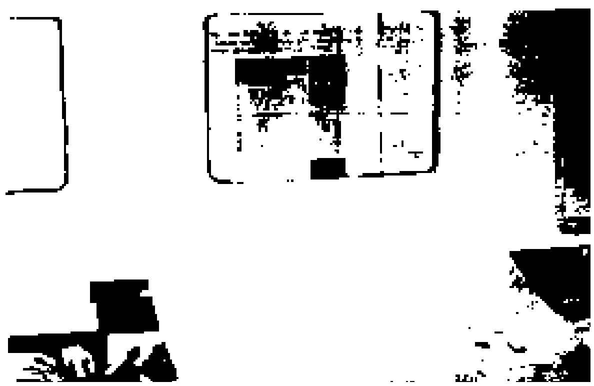

Figure 45.7. First mock-up of page make up work station.

In the above picture there is a high resolution graphic display, a control display, a tablet with a tablet menu and a mouse. Functionality of the system is simulated by making successive "drawings" of the screen. As shown in the next picture these drawings are "stored" on the wall, and "retrieved," "updated," and "changed" as the design game is played.

Figure 45.8. The wall as "store" for "interactive" display images.

This mock-up was the creative result of a major breakdown in the Utopia project, a breakdown that made us develop some new design artifacts and to shift perspective "from system descriptions to scripts for action" in participatory design. As designers we had been producing an endless number of detailed and methodologically correct system descriptions. There was just one problem. The users could not understand our system descriptions. The descriptions did not remind the users of familiar work situations. There was no meaningful role for them to play in the use of these design artifacts. The experience of using these descriptions did not relate to their work experiences. The mock-up above changed the rules of the game; it made it possible for the users to actively participate in the design process.

For example, Jon simply sat down by the mock-up and pretended that he was doing page make up work. He used the mouse, the tablet, displays and menus to crop a picture, move a headline, change a font, etc. This was done in a way that had a family resemblance to his traditional way of working. He understood the mock-up as he understood his traditional tools.

We take as an important starting point in design the idea that "in the beginning all you can understand is what you already have understood." In stating this design paradox we have been inspired by Martin Heidegger and existential phenomenology (Heidegger, 1962, and especially Winograd & Flores, 1986, and Dreyfus & Dreyfus, 1986). The point is that the mock-up did not create a breakdown in Jon's understanding. It was not present as an object in itself, but *zuhanden* (ready-to-hand) for him in his activity. Jon was primarily involved in page make up work, not in detached reflections over this activity. He was not reading or talking about a future system, but experiencing it as a *Zeug* (dress, tool, artifact) for page make up—he was literally well-equipped, rather than overloaded with equipment.

However, the mock-up is, obviously, not the same as his traditional typographical tools; hence, breakdowns in his readiness-to-hand use of the mock-up occurred. Typographic tools such as the knife became computer equipment such as the mouse and display; the mouse became a match box, the display a sheet of paper. When the spell of unhampered involvement is broken, the mock-up becomes *vorhanden* (present-at-hand) as a collection of things or objects. This is not an entirely sad story. After all, if the artifacts we use were always ready-to-hand for us, how could we then find new ways of using them? When things do not work, we shift to detached reflections of them. In the situation noted this meant reflections such as: "Is a mouse/display replacement of the typographical knife really a good design choice?" "Is the problem rather that the properties of the knife are too restricted, and that we with computer support can add some new useful properties like 'undo cut' and 'resize'?" These kinds of questions were part of the interaction between the typographer using the mock-up and a designer sitting by his side. They certainly led to new design ideas, as well as to changes of the mock-up. For example, the second version of the mock-up provided possibilities for a wider range of hands-on activities and more elaborate design language games. There were more elaborated interaction devices to try out and a more dynamic interaction with the mock-up. Aspects of the work environment and of work cooperation could be tried out.

In summary, hands-on experience is not a substitute for detached reflection. However, in participatory design it is necessary and more fundamental to support the users' ready-to-hand use of their future artifacts. Hence, an important aspect of a mock-up is its usefulness for involved activity where the users' awareness is focused on doing the task, rather than on analyzing objects and relations. Detached reflections on alternatives become part of the process when the fluent use of the typographical design tools—their readiness-to-hand—breaks down. These reflections are then grounded in a practical experience, an experience shared by users and designers in a design-by-doing language game.

Beyond the Cardboard Computer

Figure 45.9. A "second generation" UTOPIA mock-up with a back screen slide projector.

The idea of "hands on the future" as opposed to "eyes on a system description" was our main focus in the previous section. We have discussed this in terms of mock-ups built from the most simple materials, such as cardboard and paper, but there are obviously other possibilities for getting hands on the future, most notably computer-based prototypes. In this section we discuss some of the possibilities in the borderland between the "cardboard computers" and the "real" prototypes.

Second Generation Mock-Ups

As a first step let us consider some possibilities that are more complex, although still not computer based. Depending on the availability and expertise in the design group, such possibilities may include overhead and slide projectors, tape recorders, and video. These artifacts are familiar in the sense that people easily distinguish malfunction in the artifact from malfunction of the design. At the same time they provide some useful functionality beyond that which is achieved with cardboard, and they make possible a "look and feel" that is more like the future product.

Figure 45.10. Designers and potential future users envisioning the future of page make up playing with the UTOPIA mock-up.

In our second mock-up of the text- and image-processing workstation in the Utopia project we used a slide projector and a screen for back screen projection. The first impression of this mock-up was much closer to the imagined final product: The display interaction was simulated by the use of slide shows and the input devices had a "real" touch. This quality proved especially valuable when judged by people who only tried out the mock-up for very short periods of time. It was fairly easy for them to envision the future artifact by using the mock-up. This last point is important, and in many cases this alone may justify the use of slide projectors or video in a mock-up. The trade-off is that such mock-ups require more expertise and more resources, both in time and in money, and they are more difficult to change.

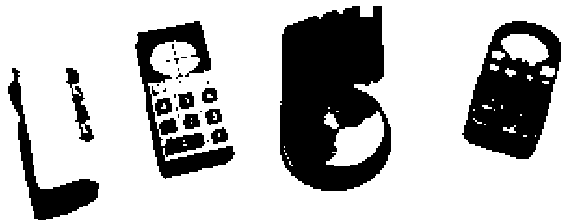

Figure 45.11. A sample of different key pad and "mice-like" input device mock-ups produced in the UTOPIA project in an attempt not to get stuck in the emerging standard interface.

Simple Mock-Ups: Advantages and Disadvantages

Before we turn our attention to the use of computers in mock-ups, we will briefly outline some advantages and shortcomings of simpler materials to get a better understanding of what might be gained or lost from the use of computers.

Until now we have looked at how to design *without* computers, not because we think that people should avoid computers in general, but because there are good reasons to think twice before using them, as well as good reasons to proceed "even" if computer support is not available. First, the mock-ups discussed so far are built with inexpensive materials. To buy expensive hardware and build advanced software early in a project may, in most situations, be directly counterproductive, especially given the possibilities of mock-ups. In other situations, however, the investments in hardware and software may not be a problem—PCs may already be massively used in the organization. Still, the use of mock-ups may pay off, because it can help generate new visions and new options for use.

Second, the characteristics of these simple tools and materials are familiar to everybody in our culture. With this type of mock-up nothing mysterious happens inside a "black box." If a picture taped to the blackboard drops to the floor everybody knows that this was due to difficulties of taping on a dusty blackboard, and not part of the design. There is no confusion between the simulation and the "real thing."

Third, such mock-ups lend themselves to collaborative modifications. The possible "operations" on the material using pens and scissors, for example, are well known to all, and with simple paper-and-cardboard mock-ups people often make modifications jointly or by taking quick turns. The physical changes are visible, and, with proper display, visible to all the participants.

However, as with any tool or technique, simple mock-ups have their limitations. Changes to a mock-up may be very time-consuming. If, for example, a different way of presenting menus is chosen, changes may have to be done to dozens of drawings, or a whole new set of slides may have to be made.

While it allows a design group to experiment without the limitations of current technology, this freedom is only a partial blessing. In the end, good design results from exploiting the technological possibilities and limitations creatively, not from ignoring them. Thus, as paradoxical as it may sound, the demand for computer knowledge in a design group using mock-ups is very high.

The simple mock-ups lack functionality: They represent physical clues with which one may create the illusion of using a future computer based artifact, but the users do have to use their imaginations along with the mock-up.

Computers in Mock-Ups: Overcoming the Disadvantages

Now let's enter the borderland between "cardboard computers" and "fully computerized prototypes." In this borderland, distinctions between the two are fuzzy. In fact, we do not see the main difference between a mock-up and a prototype as being a question of whether computers are used or not. With mock-ups—computer-supported or not—the focus is on support for overall envisionment. In a powerful analogy to film production, this kind of envisionment has been called *storyboard prototyping* (Andriole, 1989). Marty Kline, the artist who drew the storyboard for the movie *Who framed Roger Rabbit?*, makes the analogy clear: "Storyboarding is a way to look at the film without spending a lot of money. . . . It's not the ultimate film, but it represents a first chance to look at it" (Braa & Ruvik, 1989).

Moving from mock-ups and storyboard prototypes to real prototypes, the possibilities to demonstrate real computer-based functionality come into focus. Computerized prototypes differ from the use of computers in mock-ups in two important ways. We often use computers in mock-ups for purposes other than those intended for the future computer system. In the mock-up we are typically interested in using computers for envisioning the system, not to provide the real functionality of the system. Also, computers have no privileged position in relation to other materials such as cardboard and paper. They are all used on the basis of how well they contribute to creating the illusion of using the future system. In our investigations of the borderland we now consider if and how we may use computers to overcome the disadvantages described above, and if we may do so without sacrificing the advantages. We look first at the use of computers as a way to improve efficiency in building and changing mock-ups. Then we look at ways to explore technological limitations by means of computers. Finally we discuss the question of getting more functionality.

Effective Tools

The next pictures are from a recent project in which we are developing and using a computer-based hypermedia design environment that we call DesignSupport. To test and develop DesignSupport we have used it for design of a budget system. The budget system was intended to support our own research group in discussions on how to spend our funds. Most of these functions were not covered by standard budgeting and accounting systems, they were supported only by manual procedures using pen and paper, together with e-mail. The pictures are from an early mock-up/storyboard prototype of the budget system and show a scanned version of a handwritten economic overview, some links added to the drawing, and a "computer redrawn" screen image based on the handwritten overview.

After scanning the economic overview we used DesignSupport to create linked screen images. The computer-based drawing allowed us to take advantage of similarities between the pictures. For example copying shared icons between screen images was easy, and repeated changes to an image did not reduce its quality as occurs when using paper. Furthermore, several screen images could share parts. Modifications made to a part on one screen image was automatically "cascaded" to the other images.

Finally, it is worth mentioning that intermediate designs were easily saved. This encouraged the exploration of alternative designs for screen images, since backtracking to earlier versions was almost instantaneous if the current line of design proved to be unsatisfactory.

The examples of efficiency gains discussed here cover only a small sample of the potential of existing computer-based tools. Depending on the system being designed, tools such as a presentation manager or a word processor may help improve the efficiency in making and changing a mock-up.

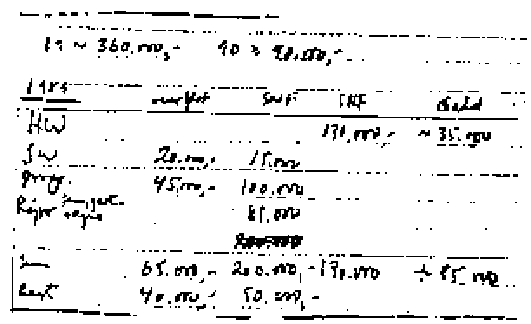

Figure 45.12. A scanned version of a handwritten economic overview

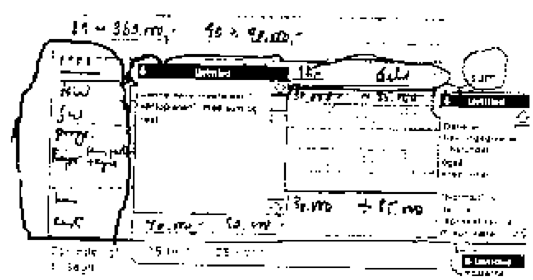

Figure 45.13. . . . some links added to the drawing

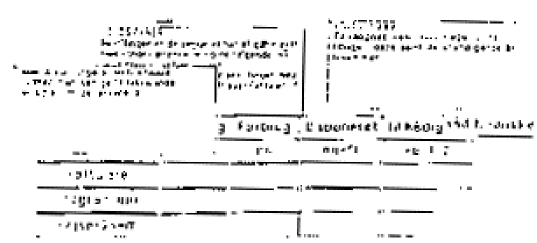

Figure 45.14. . . . and a "computer redrawn" screen image based on the handwritten overview. All the material exists in the same "hypermedia."

One should remember, however, that achieving this efficiency gain usually requires that people in the design group be skilled users of the computer-based tools, otherwise the tools may get in the way of the job to be done, changing the focus from the mock-up to the limitations of the tools.

Creating Suitable Real Limitations

As noted, the demands on computer knowledge in a design group working with simple mock-ups are very high. When a cardboard box is used instead of a desktop laser printer, someone in the group must know about such printers in order to get the game going using the box as a laser printer, not as a box. In cases where this knowledge needs to be developed, computers may be used to investigate specific technological possibilities and limitations, as illustrated by the following example.

Figure 45.15. Investigations of resolution and response times using a real computer.

During the period in the Utopia project when we used simple mock-ups we acquired a few real computer workstations with 15" bit-mapped screens. The idea was to build one or more prototypes of the emerging design. But although the hardware was powerful, the software was poor, and the prototyping could never keep up with the mock-up work. However, it was useful to be able to experiment with a 15" screen as one of our mock-up components, especially because the existing knowledge in the design group on bit-mapped screens was not comprehensive. We began to look at questions such as: "how could a newspaper page be represented with the available resolution and screen size?," "how about a spread, that is, two pages?," "how many pixels were needed to make a font readable on the screen?," and "how about using shaded boxes to represent words in small fonts?" Such questions could not easily be dealt with using the paper images and slides of our first and second generation mock-ups, but they could be investigated quite easily using the workstations with their graphic screens.

Having learned about the possibilities and limitations of our 15" bit-mapped screens, we returned to our simple paper-based mock-up to explore the possibilities of different screen sizes, such as 15", 19", and 24". We cut holes of appropriate sizes in large pieces of cardboard and placed them on the wall in front of our pictures, menus etc.

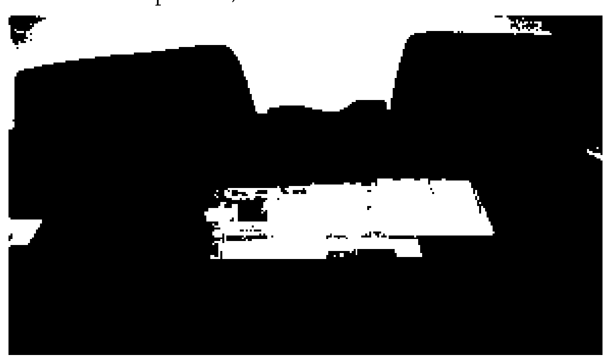

Figure 45.16. . . . and expanding the screen size with a mock-up.

More Functionality

As the third and last of the disadvantages of mock-ups we want to address by means of computers, we look at the question of functionality. This question takes us very close to the borders of "real prototypes."

Consider once more the use of DesignSupport for the creation of a mock-up of the budgeting system discussed previously. The screen images could have been printed out and used in a paper-and-cardboard mock-up like any other picture. What we did, however, was to use the computer to show a sequence of images as a slide projector does.

The next step we took was to use the "button-capabilities" of HyperCard to make it possible from every screen image to dynamically select the next image, in a way simulating how this could be done in the final system.

Using the button capabilities together with text fields in constructing the screen images made it possible to simulate a number of dynamic changes: text-entry, selection, etc. Dynamic response from the budget mock-up, such as showing the money available for inviting guests for the rest of the year, together with the estimated cost of planned and "considered" visits, was handled by a human operator simulating parts of the system. Still other kinds of response, such as sorting a list of possible conferences to attend, were not handled dynamically but rather by showing a list prepared in advance, as we did with the paper-and-cardboard mock-up.

As the example shows, there are different ways to simulate functionality, and there is also the question of what functionality to simulate in the first place. There are no simple answers, but the yardstick to apply is how the different aspects contribute to the creation of the use situation envisionment, how useful it makes the mock-up in the particular design language game. Obviously there is a tendency to implement those aspects which fit the computer best, but the tendency to implement "computer-based functionality," as opposed to different kinds of simulated functionality, is quite strong, too. In the budget system case discussed above, our programmer discovered a clever way to program Xerox-like scroll boxes in our hypermedia system. Such boxes vary in size to indicate how much of a document is shown in a window. Viewed in isolation, this approach was superior to the existing Mac-like scroll boxes of fixed size. But at the time the question of the detailed workings of the scroll bars was unimportant in relation to the creation of a suitable use situation envisionment.

In summary, computers may be used to overcome severe shortcomings in the use of simple mock-up materials. But as we shall see the costs may be high; for example, in terms of reduced possibilities for user participation. However, there is often no need to "go all the way": the best and most cost-effective envisionment may well be obtained by a mock-up from the borderland between cardboard and computers, as illustrated by the following example.

Mixing for a Better Envisionment

Our first prototype of the design environment DesignSupport was built in HyperCard and ran on Mac IIs with 21" screens. The prototype was built over a period of a few months, ending up with an almost fully functional prototype. HyperCard, however, only allowed one 9" window to be open at a time, a restriction that turned out to reduce the usefulness of the prototype dramatically. The solution to this problem was straightforward, but we were so fascinated by the prototype that it took some time to find it: To be able to show several windows in varying sizes at the same time, we simply placed printed copies where we wanted them on the big 21" screens. This worked so well that Morten immediately began to "click" on them; forgetting that it was a mock-up. He was getting an involved experience of the future use even if the functionality was missing.

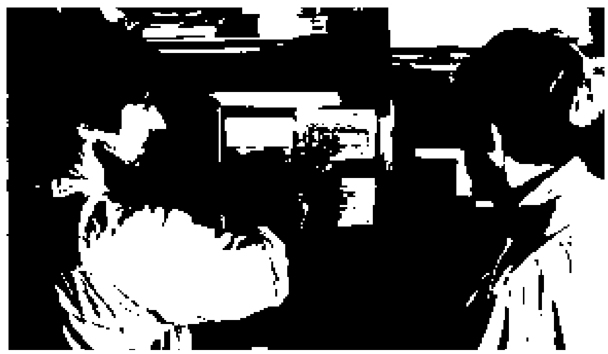

Figure 45.17. From weak prototype to strong mock-up by adding paper windows to the prototype.

This example illustrates an important difference between implementing a final system or fully functional prototype on the one hand and building a mock-up or a storyboard prototype on the other—a difference that seems to be forgotten easily once designers skilled in programming bury themselves in the computer. The point is that any design environment, computer-based or not, has limitations that at times place severe restrictions on the artifact being constructed. In the implementation of a new computer system the handling of this "tension" is a primary part of the competence of the professional designers. However, when constructing a mock-up, it is not necessary to restrain oneself to the possibilities of the computer-based aspect of the environment, unless the intention is to explore exactly these possibilities; for example, with the intention of implementing the final product using that environment.

We later implemented DesignSupport in an environment with multiple and re-sizable windows. That was a major improvement of the system as such, but our computer "blindness" for a long time prevented us from having these properties in our mock-ups and early prototypes.

Computers in Mock-Ups: Losing the Advantages?

In the beginning of this chapter we suggested that the point in using non-computer-based mock-ups was that they are cheap, understandable, and allow for hands-on experience and pleasurable engagement. Certainly, computer-based prototypes encourage hands-on experience, and in many organizations hardware and software for prototyping already exists, so resources may not be the bottle-neck. Whether it is more fun to sit by a computer or to build with cardboard, we can only guess. The remaining advantage primarily concerns the understandability of the non-computer mock-up tools and materials: How does the computer fare with this?

Unfamiliar Tools and Processes

Consider the following situation, in which computer scientists from two geographically separate groups got together to work on the design of a "shared material" supporting joint work between their two settings. They decided to use two LISP machines on a network to quickly build a computer-based storyboard mock-up. Two of the computer scientists were LISP experts; the others were less familiar with the LISP environment. Since building and modifying the mock-up was a major and integrated part of trying it out in this first session, it was the two LISP experts who operated the machines. Thus, to the rest of the group the interface to the emerging design consisted of the two LISP experts. Involvement consisted mainly of discussing ideas and their possible embodiment in the LISP machines. Several of the actions carried out by the two operators involved programming for two or three minutes. During those periods the rest of the group was inactive.

The first thing to note is that the tools and materials used in this session were not familiar to all participants. Most of them did not know what could be mocked up, and certainly they did not know how to do it. In other words they had only vague ideas about the possible moves in this design game, and they could perform just a few moves themselves. Secondly, most of the "construction work" left no visible clues; thus, the status of the mock-up was not clear to most of the participants. The result was that after a short while only the two LISP experts operating the machines were able make constructive moves. The rest of the group had nowhere to place their hands.

This could also be viewed as an example of a badly planned process. The main point in design-by-doing using mock-ups is for everyone to get hands-on experience, trying something new. This acting in the future does not happen by itself. Especially with mock-ups built using unfamiliar tools and materials, the simulated future use situation has to be carefully planned and enacted.

What's the Purpose?

As our last issue we consider the expectations of people working with a mock-up, and what the purposes of doing it are. With cardboard mock-ups it's simple: the purpose is design, and the mock-ups are used to evaluate a design, to get ideas for modifications or maybe even radical new designs, and to have a medium for collaborative changes. If experiments with computer-based mock-ups are set up in the same way and their purpose made clear, it can be equally simple. But the functional possibilities may be seductive, especially when we approach the borders of functional prototypes. Often it is possible to build a computer-based mock-up/prototype which has the look and feel of "90% of the real system," and then the use of this mock-up is interpreted by the users, or maybe even set up by the designers, in a way that presupposes that it is "90% as *useful* as the final, real system." However, this is rarely the case. When this is realized by the users their interest in using the mock-up/prototype may easily drop or disappear completely.

As mentioned earlier, successful evaluation of a mock-up requires careful planning and acting, but in addition it requires commitment from the users, resources dedicated to the purpose of evaluating the mock-up. Almost any deviation from the final system in a mock-up requires some active work on behalf of the involved (future) users. If the users are not prepared to pay this price, then using the mock-up will fail.

Mock-Ups: Prototype, "The Real Thing" or Both or ?

One of the reasons for the effectiveness of cardboard mock-ups is that nobody confuses them with the product, the future computer system; everybody knows that they do not have the functionality of a computer system. With computers in mock-ups it's different, especially when we use computers to get more functionality.

In these situations it may be difficult not to mix the appearance of the computer in a mock-up and in the imagined future product. The closer the two "roles" get, and the less familiar the computer is, the more careful one has to be in avoiding attributing the wrong aspects of the mock-up-computer to the future-product-computer.

Major Players and the Rules of the Game

Figure 45.18. Typographers, journalists, and designers in a game of soccer.

The picture above ends our story at the same place were it started—with the people missing in the first picture; showing the relationship between typographers and journalists. Here they are, in a game of soccer, and there are even some professional designers participating.

This game took place on a nice May afternoon in 1984. It was one of the activities that formed a workshop that was part of the "systems delivery" from the Utopia project. Typographers, journalists, trade union and management representatives were invited to actively participate in a three-day workshop on the design proposal from the project group. Of course, mock-ups and prototypes from the design work were tried out in hands-on sessions, but the "system requirement specification" certainly implied and included more than that. Not only the artifacts to be used were at stake; other aspects related to quality of work and product were also part of the proposal, especially questions of how work should be organized using these new tools, and what training and education the different groups should have.

On the soccer field typographers, journalists, and designers had no problem cooperating in mixed teams. This game was even more fun than playing with mock-ups. The negotiation game concerning the proposed changes of work roles and work practice was an entirely different story.

In fact, we had developed a useful work organization design kit to be used by the participants in this kind of negotiation situations, but that did not change the hard facts of reality: Some players have more power than others, and some are more vulnerable than others. For all the fun there is in design as action and in the use of mock-ups, implementation may be an entirely different game in which management prerogatives define the rules, and organizational conflicts between typographers and journalists limit forceful countermoves. In this game, often referred to as class struggle and organizational conflict, there is a temptation for the designers to think of themselves as observers just watching the game. Nothing could be more wrong in design as action, except perhaps the designers appointing themselves as referees of the game: the gods that make the other players obey the given rules.

In design as action, the rules are at stake. This is particularly true where the use of mock-ups is a way of experiencing the future. This is serious business concerning major changes of the participants' working lives. In using inexpensive mock-up tools and in establishing the pleasurable engagement of hands-on experience, the designers have to find their own role in the design game. The roles of observer and referee are not available. What defines the professional designer is the competence to find a proper role in a specific design game and to expand the space for users to participate in design as action.

References

Andriole, S. (1989). *Storyboard prototyping—A new approach to user requirement analysis.* Wellesley, Massachusetts: QED Information Sciences.

Braa, K. & Ruvik, E. (1989). *Edb-stötte til kreativt samarbeid i filmproduktion* [Computer-Support for Creative Cooperation in Film Production]. Oslo: University of Oslo, Department of Informatics.

Dreyfus, H. L. & Dreyfus, S. D. (1986). *Mind over machine—The power of human intuition and expertise in the era of the computer.* Glasgow: Basil Blackwell.

Ehn, P. (1989). *Work-oriented design of computer artifacts.* Falköping, Sweden: Arbetslivscentrum and Hillsdale, NJ: Lawrence Erlbaum Associates.

Heidegger, M. (1962). *Being and time.* New York: Harper & Row.

Kyng, M. (1988). Designing for a dollar a day. *Office: Technology and People,* 4: 157–170.

Winograd, T. & Flores, F. (1986). *Understanding computers and cognition—A new foundation for design.* Norwood, NJ: Ablex.

Wittgenstein, L. (1923). *Tractatus logico-philosophicus.* London: Kegan Paul.

Wittgenstein, L. (1953). *Philosophical investigations.* Oxford: Basil Blackwell.