Toward Expert CAD

People are remarkably flexible and robust problem solvers, whereas computers are notoriously rigid and fragile. Using insights gained through a study of human designers, we may be able to build better computer- and AI-based design tools.

David G. Ullman
Larry A. Stauffer
Dept. of Mechanical Engineering

Thomas G. Dietterich
Dept. of Computer Science
Oregon State University
Corvallis, OR

Although there are many popular textbooks on mechanical design methodology and a considerable body of work published in Europe [1–4], the mechanical design process is not well understood. Mechanical design is still difficult to teach, and there are no objective methods for determining whether someone is an expert designer.

None of the books just cited is based on empirical data. Moreover, each presents theories of how mechanical design *should* be accomplished, by encouraging a systematic design strategy based on the development and elimination of parallel solutions to a problem or subproblem. However, no experiments have been performed to determine whether these methods are really effective in producing good designs.

There are other goals that an empirical study of the mechanical design process could meet. It has been conjectured that manufacturing productivity has increased many fold since the turn of the century, but that design productivity has increased only slightly over the same period. The insights gained from such a study could help in improving the efficiency of the design process and the quality of resulting designs [5].

This work is especially relevant as artificial intelligence methods are incorporated in CAD tools. AI products have the potential to assist designers in phases of the design process when existing CAD tools are inapplicable. For example, AI techniques may be useful during the conceptual design phase—a time when crucial design decisions are made.

Another possible application of AI is to the problem of recording and replaying design history. Now, the primary record of a design is the set of final drawings produced by the engineer. The various alternatives, trade-offs, and design constraints that contributed to the final design are not recorded. Without these data, a design must be reconstructed from scratch when it is modified, or when a similar device is designed.

Finally, any CAD tool must provide a natural conceptual interface to the designer. If an engineer cannot understand what a CAD system is doing or how it is approaching a problem, he or she is unlikely to make effective use of it. Similarly, if an AI-based CAD system cannot understand what the engineer wants to do, it will be unable to provide assistance or record the decision-making processes. In short, intelligent CAD tools must employ human-like methods if they are to be useful.

Design Problem

At Oregon State University, we developed two problems with several goals in mind. First, we wanted to observe all phases of the design process. As a result, we provided our subjects with incomplete, high level specifications, and

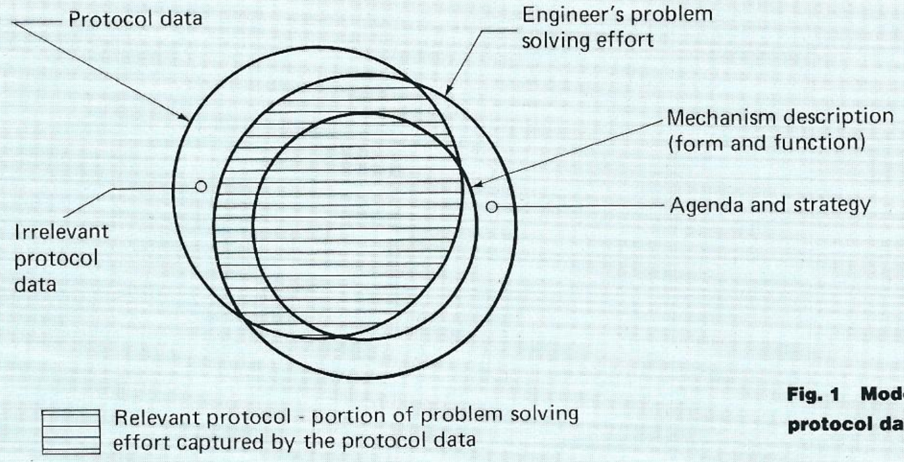


Fig. 1 Model of relevant protocol data.

followed their progress until they produced detailed working drawings for at least some parts of the final design.

Second, we wanted to explore the difference between product designs and one-off designs, because the constraints on the problem of designing for a product are much different from those for designing a one-of-a-kind device.

Third, we wished to investigate the relationship between the engineer's knowledge and skills, and the requirements of the problem. For this, we took data from graduate students with limited design experience, as well as from experienced mechanical designers. We also selected problems for which our experienced subjects could be expected to have a high degree of expertise.

One-off design. Appendix I presents the two design problems we used. One, which we called the "flipper-dipper" and was based on a consulting contract completed by one of the authors, involved designing a machine to grasp and position a thin aluminum plate onto the surface of a water bath. The machine had to dip both sides of the plate, one at a time.

The problem required a simple knowledge of kinematics and some actuation technology, such as manual manipulation, pneumatics, or small electro-mechanical transducers. Only three of these machines were to be constructed.

Product design. The second problem, which was developed with the cooperation of a major computer manufacturer, was to design the contacts and compartment for batteries for installation by a robot in a small portable computer. The problem required knowledge of metal springs, molded plastic materials, and robot assembly constraints. Over the expected lifetime of the product, approximately 1.8 million units would be produced.

Method

We used a technique called protocol analysis (see box), which involves presenting a subject with a problem and asking him or her to "think aloud" while solving it [6]. The subject's verbal "protocol" is then analyzed to obtain a "trace" of his or her problem solving steps. Through careful analysis, it is possible to develop a coherent explanation of the problem solving methods employed by the subject. (A detailed discussion of the methodology is given in the box.)

Analyzing Data

In evaluating the protocols, it is important to keep in mind that the protocols give a verbalization of a portion of a subject's short-term memory during the design procedure [7]. Since only one thing can be verbalized at a time, the subject cannot verbalize all that he or she is thinking.

Moreover, all verbalization is not necessarily pertinent to the design. In Figure 1, we show the relationship between the engineer's problem solving effort and the protocol itself. Some of the information is irrelevant and can be eliminated. What is left is the relevant protocol, which captures a portion of the subject's problem solving effort.

This contains not only a description of the form and function of the mechanism being designed, but also the designer's strategy and agenda. Statements such as "I think I'll look at a side view first" do not correspond to any change in the evolving mechanism, but certainly indicate the subject's design strategy.

As a result, we see the subject as having two basic types of thoughts: those pertaining to the form and function of the mechanism being designed, and those referring to the problem solving process itself. With the protocol analysis method, we have captured part, but not all, of the form, function, and process via the subject's verbalizations, which are represented by the hatched region in Figure 1. The remainder of the design process must be inferred, but, as shown below, careful analysis of the protocol can give meaningful results with minimal inference.

Breakdowns. The first step in analyzing the protocols was to develop a coarse breakdown, which was done by simultaneously watching the videotape and reading the transcript. We did this to understand the flow of the protocol, to familiarize ourselves with the design, and to identify interesting areas for further analysis.

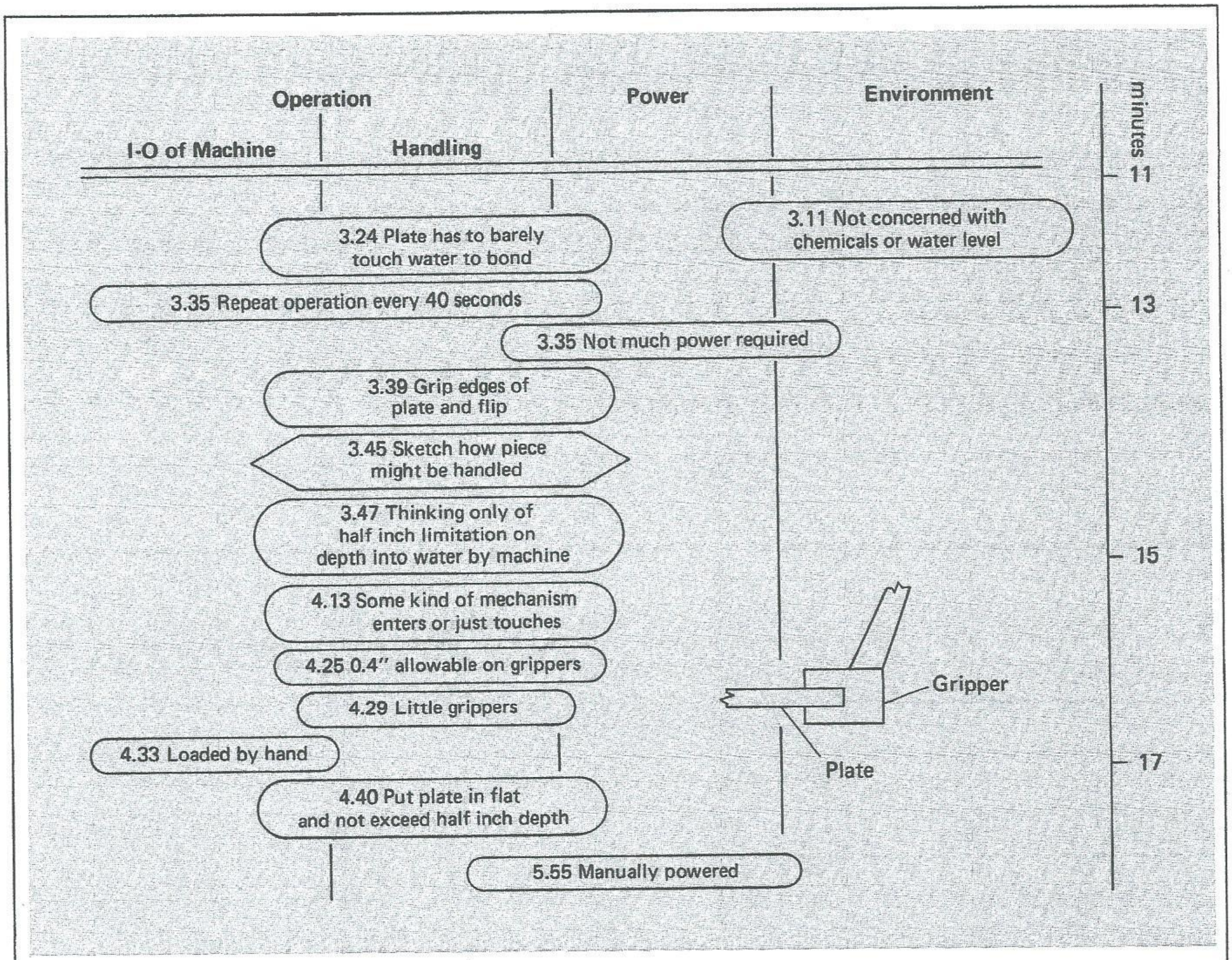


Fig. 2 S6's functional coarse breakdown.

Coarse breakdowns. Here, the protocol was analyzed to find points where the subject articulated specifications, constraints, design considerations, and design decisions. Also included were points where drawings were constructed or calculations were made.

Because the goal was to identify the overall flow of design decisions, the coarse breakdown omitted many details surrounding each of the subject's decisions. Typically, only 20 percent of the subject's utterances were included here.

Our procedure was to have two researchers independently prepare coarse breakdowns, then compare and combine their results for a final version. With practice, we could obtain 75 percent agreement.

We tried various notations for capturing both the chronology of the design and the development of form and function. Many researchers have used a tree to represent the design process [4, 7–8], but a standalone tree of a design's final form or function cannot sufficiently convey the chronology and the complex interrelationships of design decisions.

A coarse breakdown for subject S6, an experienced engineer solving the "flipper-dipper" problem, is shown in Figure 2. The three main headings there correspond to the three basic functions that warranted consideration (as conceived by this subject): *operation*, *power*, and *environment*. The vertical axis represents time, and advances down the page.

The protocol is divided into a sequence of events. In each, the subject is focused on some entity: a *form*, a *function*, or a *strategy* for carrying out the problem solving process. Each event is represented by a block. Blocks with rounded ends indicate events in which the subject worked on some function, while rectangular blocks show form events. Blocks with triangular ends designate events that dealt with the strategy of the problem solving process or with the identification of new *agenda* items for consideration. The numbering of the blocks gives the page and line number in the transcript where the event appears and, thus, the sequence of events.

Getting started. After reading the problem and gaining an understanding of what was required, S6 began to think in terms of the three main functions. The operation function was subdivided into two secondary ones: *input and output of the machine* and *handling of the plates*.

The subject had no form for the machine in mind at this time, but was concerned with its functions. After addressing

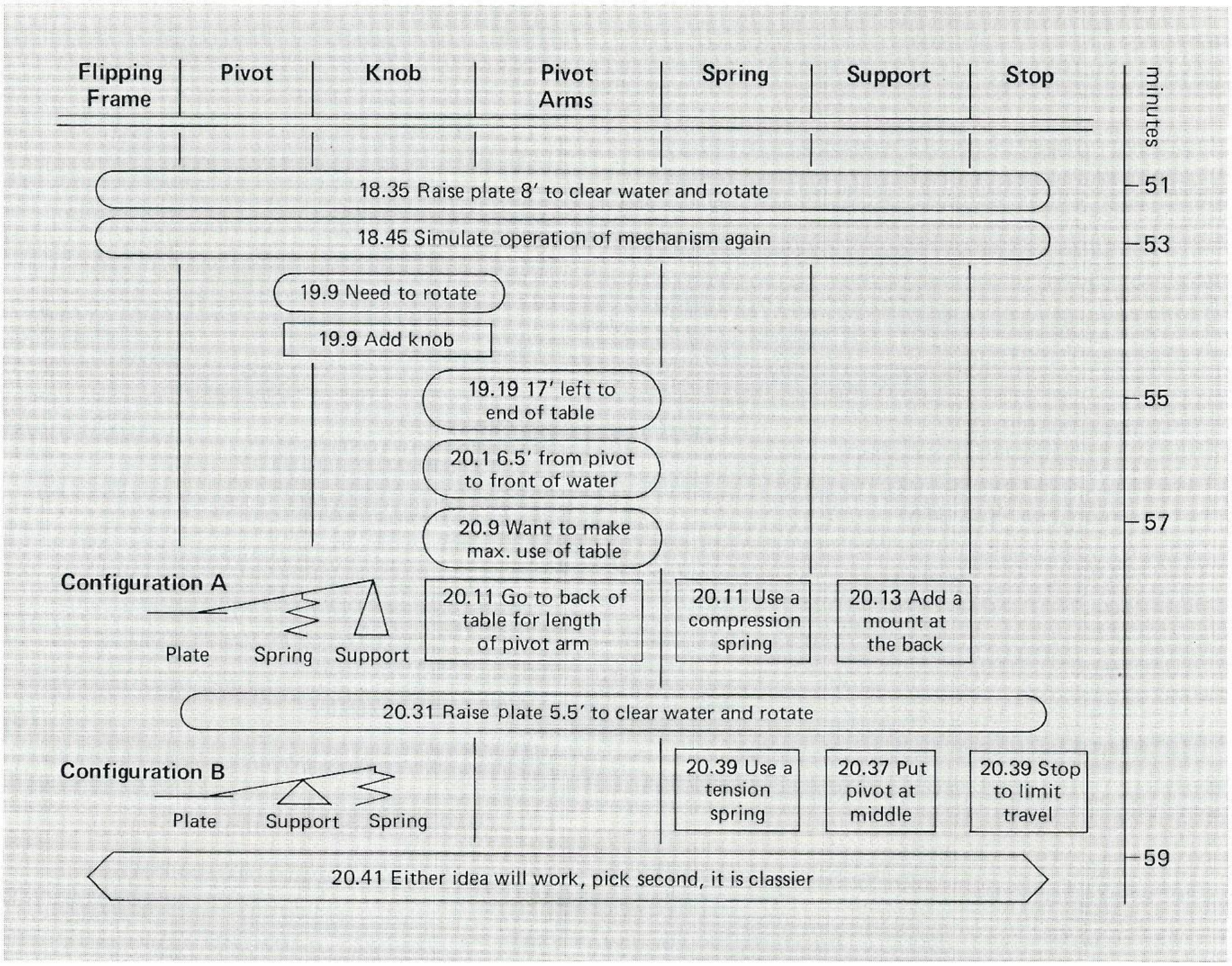


Fig. 3 S6's coarse breakdown: form selection.

each of the main functions, he focused on plate handling. In event 3.45, he began to develop his first form.

He went on to talk vaguely in event 4.13 about "some kind of mechanism" for handling the plates, and eventually, in event 4.29, sketched his first form, which he called "little grippers." These were still quite conceptual and nebulous. After this brief functionally-oriented thought process (shown in Figure 2), the subject sketched his first conceptual design.

After this, S6 never considered other ideas. His original design remained and evolved into a final detailed design. After the first conceptual design was developed, the subject generally organized his thoughts around individual components (forms), not around the functions shown in Figure 2.

To reflect this change in focus from the first conceptual design, we have rearranged the columns of the coarse breakdown so that they denote forms instead of functions (see Figure 3). In this form-oriented scheme, a function sometimes spans two or more columns when it includes more than one form (e.g., 18.45 and 20.31).

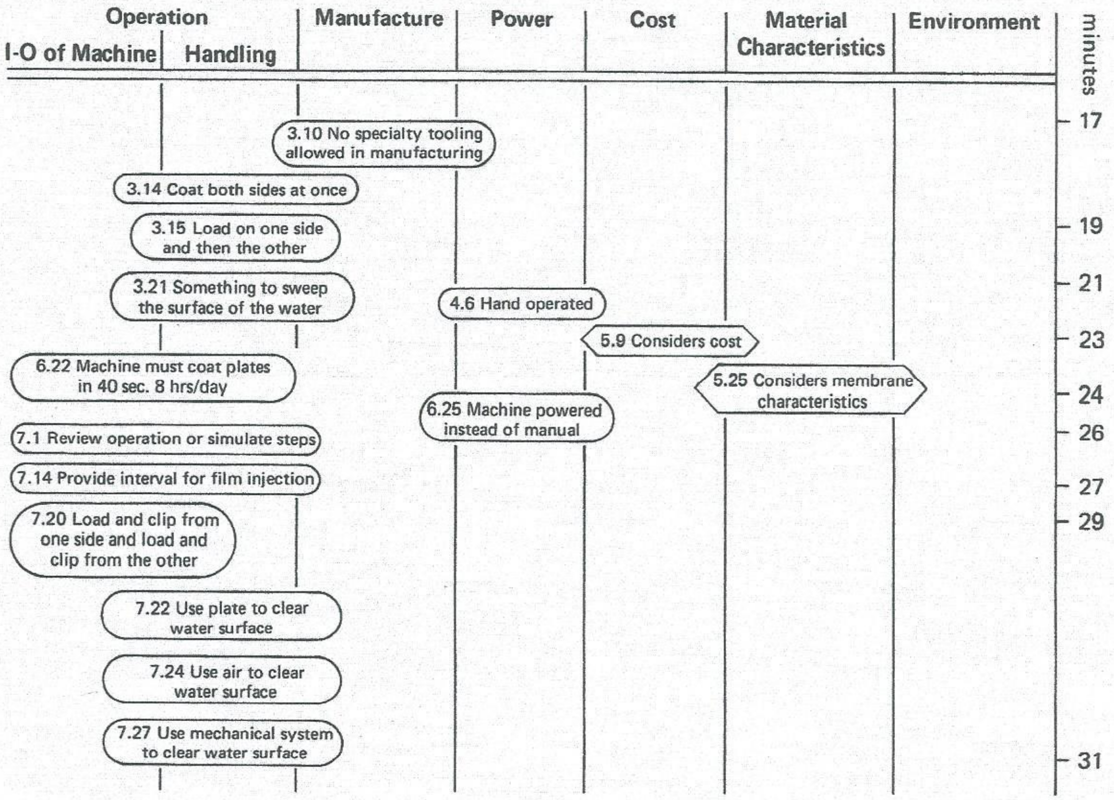
(A form block can also span two or more columns if it later evolved into different forms. However, this was not seen in the example, and rarely happened in any of the protocols.)

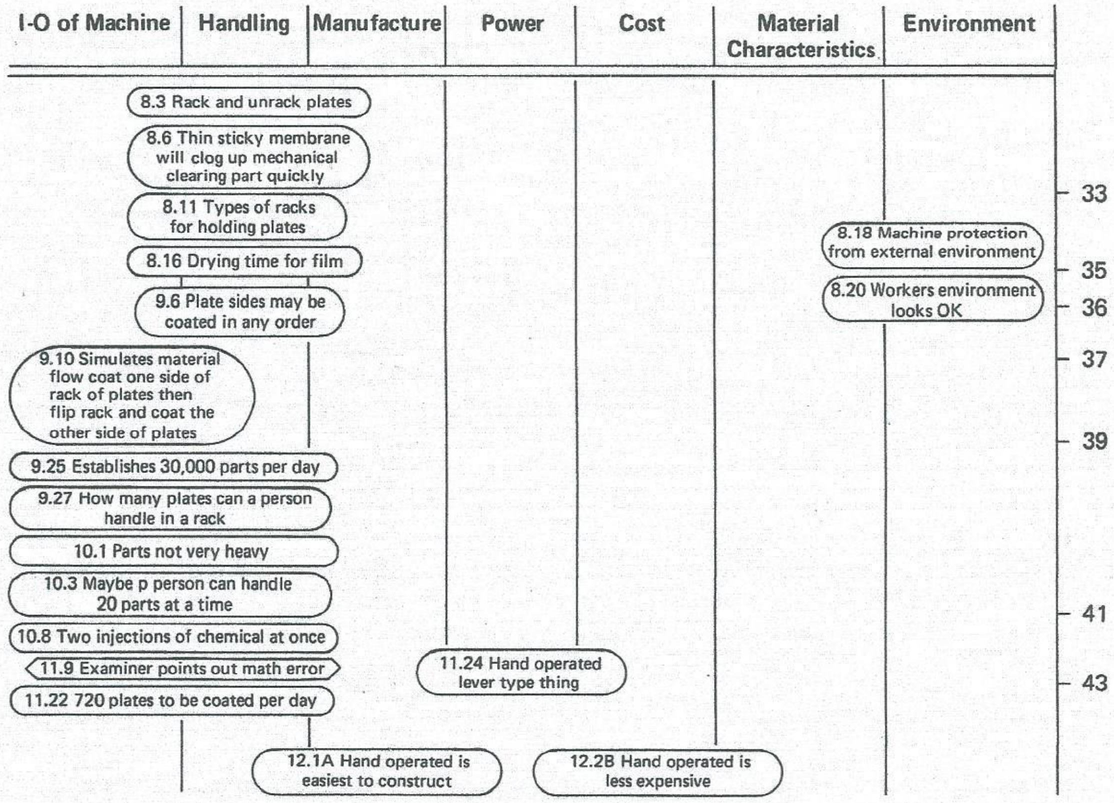
Here, S6 is still finalizing the relationship between the forms. In the beginning in Figure 3, the subject stated a function that will eventually affect several forms (event 18.35): "[I]t would only take an altitude of about eight inches to clear the water and to rotate the part, then come down and register on the side of the tank. . . . " He then refined preexisting forms to achieve this. Configuration A in Figure 3 is a graphical representation of the mechanism at this point.

Next, he restated the need to raise the plate to clear the water bath and rotate it, but this time, in event 20.31, he calculated a distance of 5½ -in. instead of 8-in., as in event 18.35. In actuality, the 5½-in. distance would cause physical interference in the machine's operation. The erroneous constraint was established as a new function.

The subject then generated Configuration B in Figure 3, which basically switches the positions of the support and the spring. It appears that the 5½-in. constraint served as a trigger for this second design.

Rather than analyze both configurations in greater detail, the subject simply chose the second idea over the first, for apparently minor reasons, in event 20.41. He never again used the 5½-in. distance for raising the plate, and eventually designed a machine that raised the plate 8-in. above the water bath.


Operation
I-O of Machine
Handling
Manufacture
Power
Cost
Material Characteristics
Environment
minutes
17
3.10 No specialty tooling allowed in manufacturing
3.14 Coat both sides at once
19
3.15 Load on one side and then the other
21
3.21 Something to sweep the surface of the water
4.6 Hand operated
23
5.9 Considers cost
6.22 Machine must coat plates in 40 sec. 8 hrs/day
5.25 Considers membrane characteristics
24
6.25 Machine powered instead of manual
26
7.1 Review operation or simulate steps
7.14 Provide interval for film injection
27
29
7.20 Load and clip from one side and load and clip from the other
7.22 Use plate to clear water surface
7.24 Use air to clear water surface
7.27 Use mechanical system to clear water surface
31


I-O of Machine
Handling
Manufacture
Power
Cost
Material Characteristics
Environment
8.3 Rack and unrack plates
8.6 Thin sticky membrane will clog up mechanical clearing part quickly
33
8.11 Types of racks for holding plates
8.18 Machine protection from external environment
8.16 Drying time for film
35
8.20 Workers environment looks OK
9.6 Plate sides may be coated in any order
36
37
9.10 Simulates material flow coat one side of rack of plates then flip rack and coat the other side of plates
39
9.25 Establishes 30,000 parts per day
9.27 How many plates can a person handle in a rack
10.1 Parts not very heavy
10.3 Maybe p person can handle 20 parts at a time
41
10.8 Two injections of chemical at once
11.9 Examiner points out math error
11.24 Hand operated lever type thing
43
11.22 720 plates to be coated per day
12.1A Hand operated is easiest to construct
12.2B Hand operated is less expensive

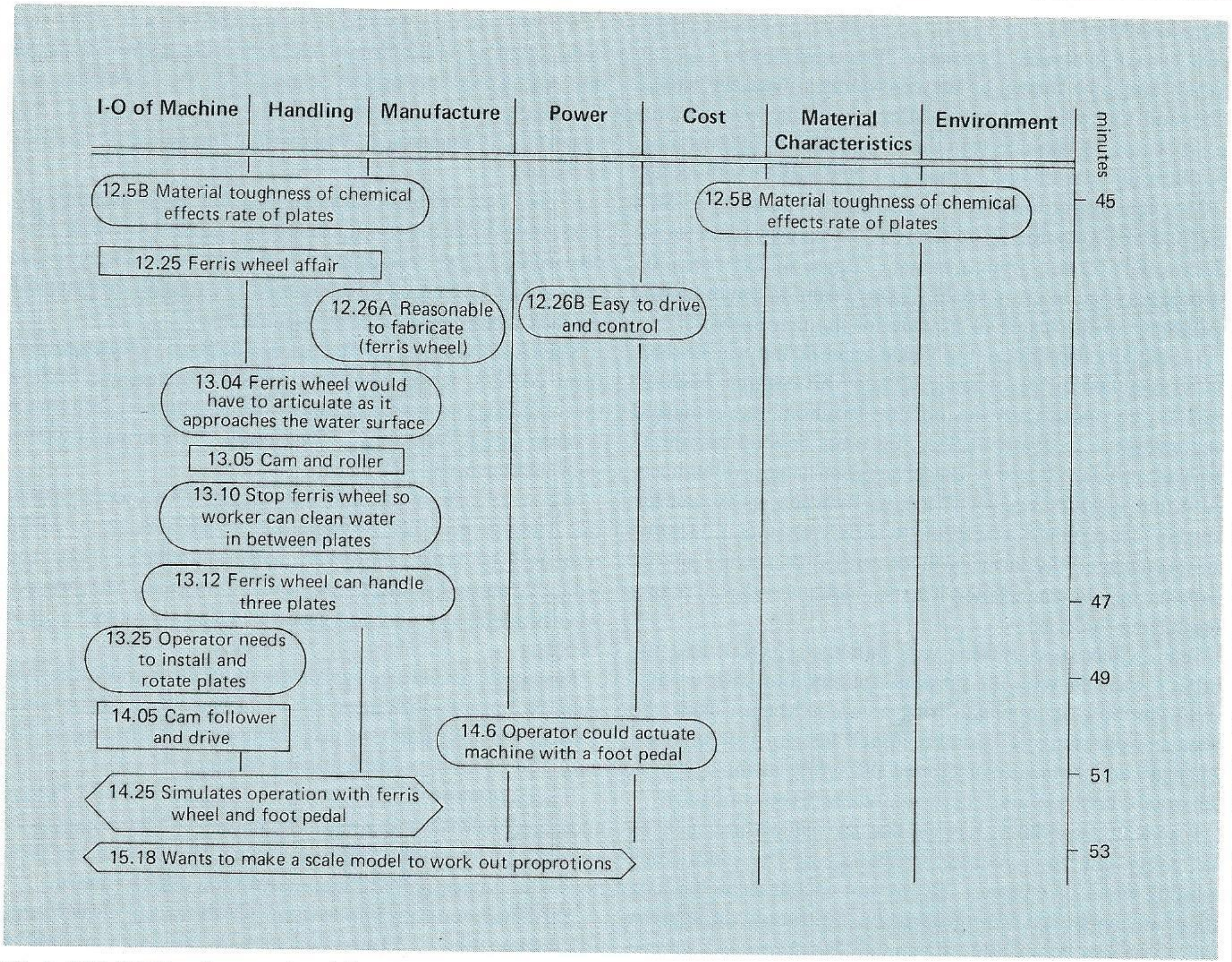


Fig. 4 S5's functional coarse breakdown.

Generating concepts. The protocol for subject S5, another expert engineer working on the same problem, was analyzed in the same manner (see Figure 4). While S6 arrived at his first conceptual design in 20 minutes, S5 required 42 minutes, largely because he considered not only "operation," "power," and "environment," but also "manufacturing," "cost," and "material characteristics" before developing a first conceptual design and concentrating on the operation of the machine.

At this stage, the subject also introduced some peripheral functions that we never intended him to address. Two of these were how to clean excess chemical from the water bath (events 3.21, 7.22, 7.24, and 7.27), and how to handle the plates after coating (event 8.3 and 8.11). Both of these problems and several other minor concerns would be important in actual design, but the examiner told the subject not to worry about them in order to keep the problem solving sessions to a manageable length.

One point consumed nearly five minutes of the protocol, when S5 became concerned about the number of plates that had to be handled in a day. In event 7.25, he calculated a quantity of 30,000 plates per day. This led him to consider such elaborate ideas as making the chemical coating a continuous process or coating several plates at a time.

Eventually, the examiner pointed out a math error, and the quantity was recalculated at 720 parts per day, in event 11.22. The subject then realized the problem would not require such a complex solution, and proceeded.

Even though the large quantity requirement was the result of a math error and the subject knew it, a machine was designed that could handle three plates at a time to unnecessarily speed up the process. In event 12.25, the subject stated, " . . . the idea just came across my mind to use, kind of like a ferris-wheel affair . . . ," which would make the dipping process semi-continuous.

This and other examples suggest that the subject never forgot the result of his math error. After he mentioned the ferris-wheel type process, he focused on developing the idea. For the next eight minutes, he imagined how the operation of a ferris wheel could be applied to his problem and, in event 13.05, added forms such as a cam and roller to cause the plate to articulate as it approached the water surface. By event 15.18, the subject arrived at a concept, and his thinking became more form-oriented.

Fine breakdowns. Interesting sections of the coarse breakdowns were analyzed to produce "fine breakdowns," which follow the same format as the coarse ones, with the exception that the subject's every utterance is analyzed. While most of the information is in the protocol, some things must be inferred for a full understanding of the subject's thinking.

The fine breakdown for a 1.5 minute section of S6's protocol is shown in Figure 5. Here, S6 was trying to locate the support for the pivot arms (see Figure 6). He began by stating his goal, in event 66.23: "[Do] I want to put them [the supports] on the inside or outside [of the pivot arms]?"

He initially considered the inside location but was unsure, so he sketched the pivot arms to "see" his options. In event 66.27B, he considered the outside location, but remained unconvinced. In event 66.33A, he went beyond his stated options and considered a forked support—a sort of compromise that supports the pivot arm on both the inside and the outside. Thus, while attempting to satisfy his goal of locating the support of the pivot arm on either the inside or the outside, the subject relaxed it to the higher-level goal of just supporting the arm.

The forked support idea was quickly rejected, and the subject was left with the original options, but with no new ideas on how to evaluate and compare them. Then, S6 became opportunistic. He abandoned his generate and test strategy, and instead investigated the safety of the design in the area where the pivot arm and support are connected.

It appears from the videotape that the subject had focused on safety due to a mental simulation of the pivot arms rotating about the support. He did not appear to be thinking about any particular support location, but simply the support and pivot arm connection in general. He addressed several safety concerns in events 66.37-66.45A. Since he did not evaluate the inside or outside location with respect to safety, he was presumably thinking about them collectively.

By event 66.45B, he had not identified any important safety concerns, and simply dropped the topic, evidently convinced there was no safety problem. He returned to the support location problem and, after a few seconds of thought, recognized that an outside location for the support would make it easier to screw the support to the table than an inside location (event 66.47). He therefore decided on the outside location, and drew the support on his sketch.

How Designers Work

The following discussion is based primarily on the partial reduction of three protocols (S2, S5, and S6)—all from experienced mechanical designers. We are confident that the findings below are common to experienced mechanical designers. We do not, however, know whether they are examples of good or bad design practice.

Designers pursue a single conceptual design. In all of the protocols, an initial, preliminary design was established very early. This design became the theme for the solution to the problem, and was modified and patched until it worked. There is little evidence that alternative conceptual designs were considered.

We seldom saw parallel development of more than one idea, and if several alternatives were initially developed, all but one would be rapidly eliminated. We observed this single-concept strategy not only for the overall design, but also for individual components within the design at all levels of detail.

If problems with the original concept were later uncovered, they were solved by patching the design rather than discarding it and developing a new approach. The first idea was almost sacred, and sometimes even highly implausible patches would be applied to make it work.

This is contrary to the principles of good design practice from current design theories [3–4]. However, we believe that this single-concept strategy is quite common, and that it is often suppressed by working groups or by design reviews with management.

We were concerned that our subjects were behaving in this manner possibly because the design was never to be reviewed or built. However, this is unlikely, since the subjects were also aware that they were being videotaped. This probably encouraged them to take extra care in their design.

Moreover, experience in protocol analysis indicates that subjects tend to work more closely and carefully on camera. Hence, we conclude that this single-concept strategy is the subjects' normal problem solving strategy.

It is interesting that the protocol studies of software design (see box) have found similar behavior in their subjects. In virtually all cases, the design developed around a kernel idea that was successively refined during the design process.

This has important implications for the study of mechanical design theory. Either our subjects are poor designers because they employ a weak strategy, or else the theories about efficient design methods are incorrect. Since the subjects are experienced designers and well-respected by their employers, there is little reason to believe that they are inefficient, or that they produce inferior designs.

However, it is conceivable that design is such a complex task that there are very few people who do it well. In this case, the methodologies may be correct, and our subjects may need retraining or better design tools. The only way to settle this issue is to develop some means for judging the efficiency of the design process and the quality of the designs. At present, we do not have objective ways of making this judgment.

It is important to resolve this question before we construct AI-based CAD tools for designers. If the single-concept strategy is appropriate, it will significantly simplify the development of CAD tools, because they will not need to store and update multiple alternative designs.

On the other hand, if the single-concept strategy is inferior, it will be critically important for AI-based CAD tools to support multiple alternatives and to assist the designer in managing them. This will require much more complex software. The complexity of such systems suggests that human designers follow the single-concept strategy because it is cognitively easier, rather than because it produces superior designs more efficiently.

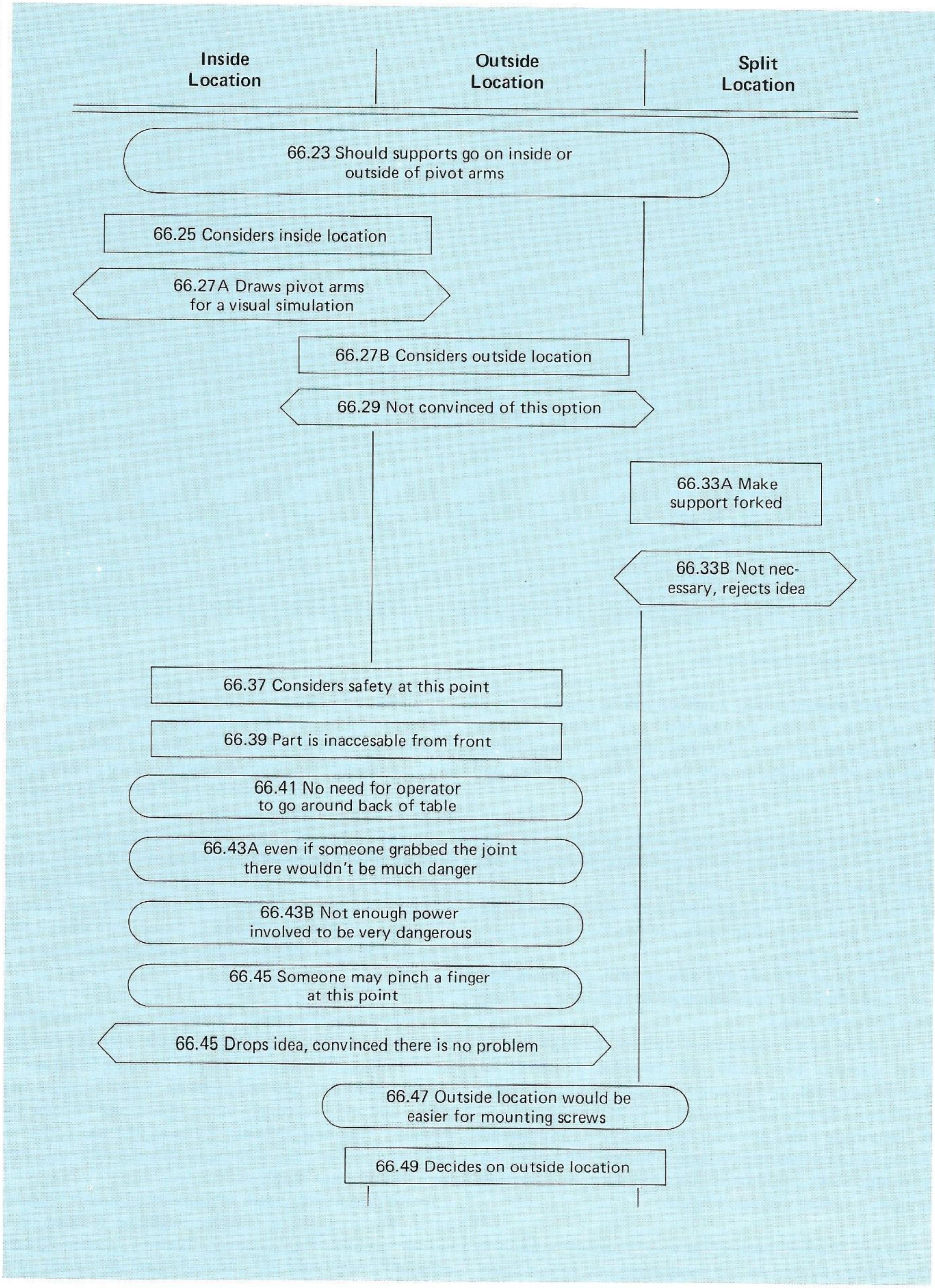


Fig. 5 S6's fine breakdown.

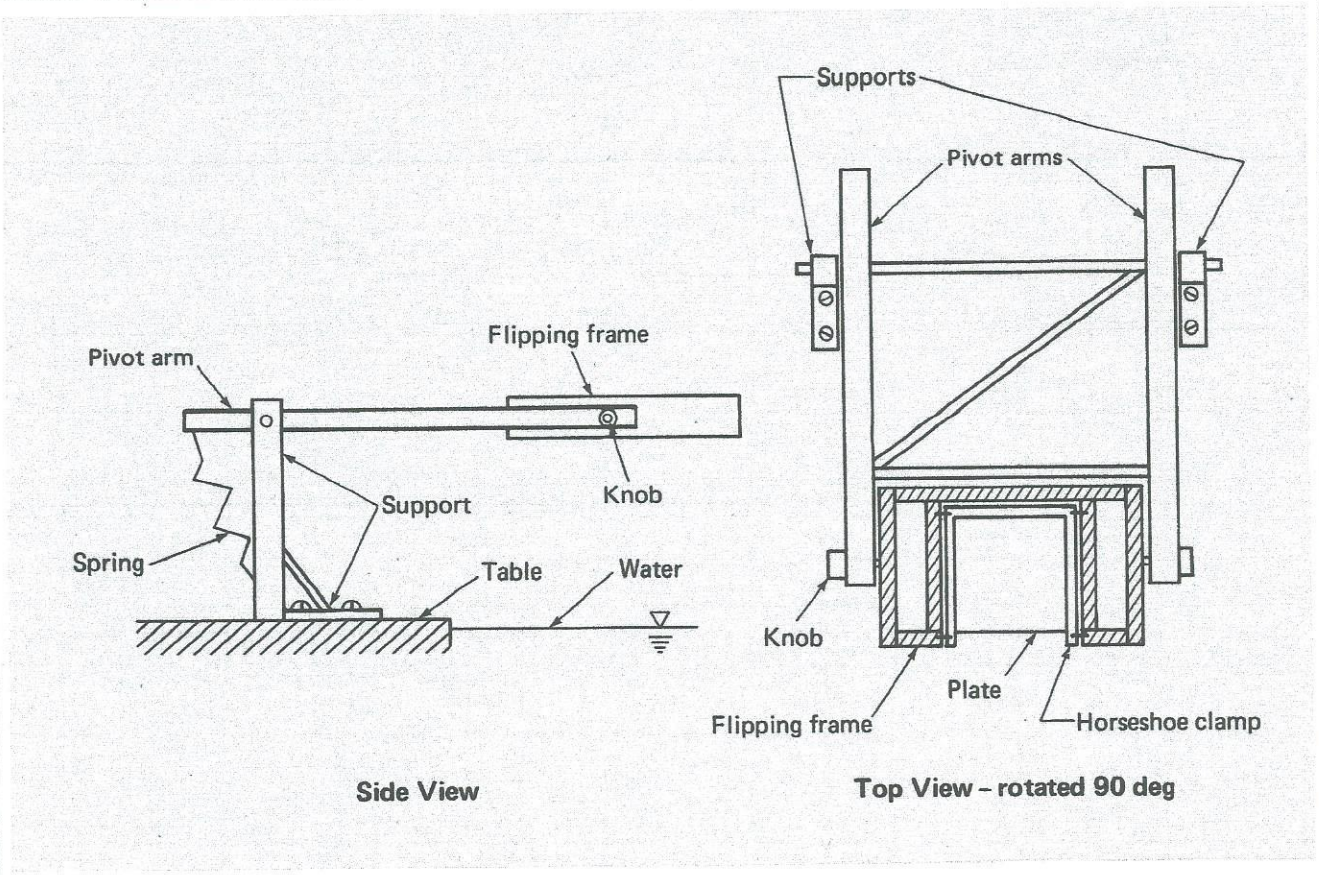


Fig. 6 S6's flipper-dipper design.

Notes and drawings play a critical role in design. During the problem solution, the engineers took extensive notes, both mental and written; most often, they were in the form of drawings and sketches. We observed six uses of drawing and sketching in the design process:

- To archive the geometric form of the design.
- To communicate ideas from one designer to another, and from designers to manufacturing personnel.
- To provide a visual simulation of ideas. (Often, the designers sketched various options to simulate configuration or information flow.)
- To act as an analysis tool. (Frequently, missing dimensions and tolerances were calculated directly on the drawing as it was developed.)
- To serve as a completeness checker. (As sketches or other drawings were made, the remaining design details become apparent to the designer. This, in effect, helped establish an agenda of design tasks left to accomplish.)
- To provide a kind of "external memory." (The designers often made sketches to help them remember ideas that they were afraid they might forget. This was not always successful, and is further discussed below.)

The use of notes is consistent with the behavior found in the software studies. In our comparison, we cannot determine whether more notes are taken by subjects working in unfamiliar domains as reported by Adelson and Soloway [9].

It is interesting that in the software studies, the process of mental simulation and test-case execution served many of the same purposes as drawing does in mechanical design. In particular, mental simulation of algorithms and programs has been observed to provide the first and last of the functions listed above in software design.

Current mechanical engineering CAD tools only serve the designer directly on the first two items, and indirectly on the next two. CAD tools do not offer any active mechanisms for pointing out missing dimensions or assisting in calculations. It is in the development of tools to better serve the last two items that we may see an increase in design efficiency.

Designers progress from systematic to opportunistic behavior. Initially, each designer went over the problem statement to get an overview of what he or she faced, and to identify important design functions. Subjects noted key points and established an initial agenda for solving the problem. This was followed by a systematic period of conceptual design. The subjects usually followed an organized plan of attack at this early stage.

As the design progressed, subjects became more opportunistic. For example, a designer might be focused on a specific problem and suddenly "notice" another (usually adjacent) problem that either overwhelmed the current focus or displaced the other items on the subject's informal agenda. We saw an example of this when S6 interrupted his efforts to locate the pivot arm supports and shifted to safety issues (Figure 5).

In this case, we believe that S6 switched his attention due to a mental simulation of the pivot arms rotating about the support. In other areas, we suspect that once the basic form has been developed and sketched, the designer starts to be driven by visual cues. Ideas and problems are triggered by noticing patterns or configurations in the sketches.

Sometimes, the focus of attention is immediately shifted to one of these, perhaps because the designer doesn't want to lose the "good idea" by forgetting it. We also suspect that as the design progresses, the complexity increases to the point where the designer can only keep a small portion of the whole design in short term memory or in front of him or her in the form of a sketch. Consequently, an agenda is formed by what he or she can remember and see.

There is no reported evidence of this behavior with the software designers. Both our work and the software studies show the flow of the design from the abstract to the concrete, but this flow from systematic to opportunistic was unexpected. Perhaps we see this behavior because our design problems are significantly larger than those studied previously, and our subjects reach the point where they become cognitively overloaded.

There is no discussion of this behavior by the design theorists, either. The design theories are, by definition, systematic strategies for conducting design, and thus our designers again diverged from the current thinking on what makes for a good design process.

There are two implications of this behavior for the development of future CAD systems. First, CAD systems should provide support for opportunism. They should not put the designer in a methodological straitjacket that prevents him or her from pursuing important opportunities and problems as they are discovered.

Second, if our speculations are correct that one cause of opportunism is cognitive overload, AI-based CAD systems should also assist the designer in managing the complexity of the design. An AI-based system, for example, could maintain an agenda of suspended tasks and remind the designer of tasks requiring further attention.

Designers do not always keep their designs balanced. In most design theory and in the software studies, it is claimed that the designer makes an effort to keep the design balanced. We see both supporting and opposing evidence for this.

Subject S2, working on the battery contacts problem, seemed to focus on one small part of the total problem before addressing many other aspects. This one aspect was developed in great detail before the subject continued.

Subject S6, working on the flipper-dipper problem, seemed to operate in a balanced way. He repeatedly worked on each component in the design, beginning with the clamp that holds the plate and ending with the question of how the machine attaches to the table.

This corresponded to a spatial traversal of the device from one end to the other. At the end of each pass, the subject analyzed the overall design, then returned to the clamp and addressed each component again. This was his overall strategy, although he did diverge from it to make minor decisions. Only when S6 got to the detail stages of the design did he concentrate on developing one component at a time.

We hesitate to draw any conclusions from these observations because of the ambiguity of the term "balanced development." We define it as the effort to keep all the elements of a design at the same level of abstraction while moving the design toward completion.

However, an alternative definition is that the design process is balanced if the designer addresses every element of the design on each pass, regardless of the level of abstraction of the elements. According to this definition, each pass is intended to reduce the level of abstraction of the design as a whole.

In our protocol analysis, it is easy to determine whether the subject is satisfying the second definition of balanced development. However, to evaluate the first, we must be able to determine when elements are "at the same level of abstraction." We do not yet have an objective definition of levels of abstraction.

As with the other behaviors already discussed, any departure from balanced development suggests that future CAD tools should allow the designer great flexibility in exploring the design space. Tong's work on AI-based CAD tools for VLSI design attempts to provide this flexibility [10].

Tong suggests that departures from balanced development are necessary and appropriate when the design problem contains identifiable bottlenecks. In such cases, the designer should take time to explore the bottleneck in detail and determine whether a feasible design can be constructed. This may explain the behavior of S2, who was working on the battery contacts problem which has a very tight spatial constraint—a bottleneck, in Tong's terminology.

Designers are forgetful. We have seen repeated evidence that the designers forgot their earlier decisions.

For example, early in his design, subject S6 considered the problem of how to mount the support for the pivot arms. He decided to mount the support on the outside, and even made a sketch recording the decision. Later on, he reconsidered the problem; again he considered alternatives, rejected some of them, and settled on the "outside" location. Throughout the second episode, he gave no indication that he remembered his previous solution.

There are cases, of course, in which our subjects readdressed problems because new constraints or interactions had arisen and required attention. However, this cannot explain cases such as S6's, in which the same decision was made twice, with no new constraints or any sign of recollection on the part of the subject.

This is not at all surprising given the complexity of the design process. Future CAD tools should provide support for recording design decisions and helping the designer keep track of what work has been accomplished and what remains to be done.

Continued on page 70

Appendix I

Design Problems

Flipper-Dipper

Our manufacturing company needs a machine to coat thin aluminum "plates" with a thin chemical layer cast on the surface of a water bath. The machine will dip the plates into this chemical bath. We will produce three of the machines, and have standard machine tools in-house. The machine must coat the plates as follows:

- A worker loads the machine with an 0.063-in. × 10-in. × 10-in. aluminum plate. Since the worker will load and unload these plates all day from a standing position, fatigue should be kept to a minimum.
- The worker visually ensures that the surface of the water is clean, and then uses a syringe to inject a pre-measured amount of chemical in solvent solution on the surface of the water, where it spreads as an oil slick over the surface. When the solvent evaporates after just a few seconds, the 500 Angstrom thick chemical layer is ready for application to the surface of the fin. The chemical is nontoxic and safe to handle.
- The chemical is applied to the surface of the plate by gently lowering the fin onto the water, where surface tensions cause them to bond instantly. Once the plate is coated, it is moved away from the surface with one edge leading, and the process is finished for that side of the plate. The excess chemical on the surface of the water is cleared from the bath manually by the worker (the layer is very thin and sticky).
- The process is repeated to coat the other side of the plate. After coating both sides, the plate is then presented to the worker for unloading. Dipping the plate, adding new chemical, flipping the plate, and dipping it again takes about 40 sec.

These are the constraints on the problem:

- The plates can only be handled by the edges. Only the edge 0.25-in. around the periphery of the fin can be touched by either the worker or the machine at any time. See Figure 7.
- The water must be kept clean, since any impurities can affect the integrity of the chemical. This is especially true of organic materials.
- Parts of the machine that hold the plate can enter the water, outside the periphery of the plate, to a depth of up to 0.5-in.
- It is anticipated that the machine will mount on the table surface in the areas shown. The machines cannot extend beyond the boundary of the table.
- The water bath level is automatically maintained 0.5-in. below the surface of the table, ±0.003-in.

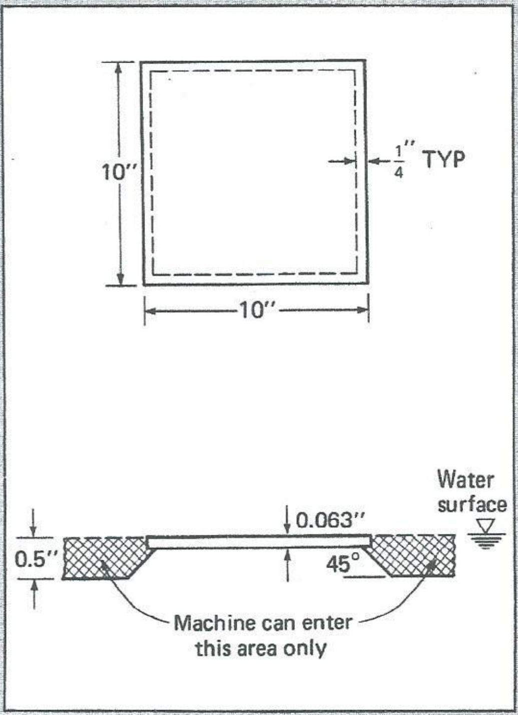
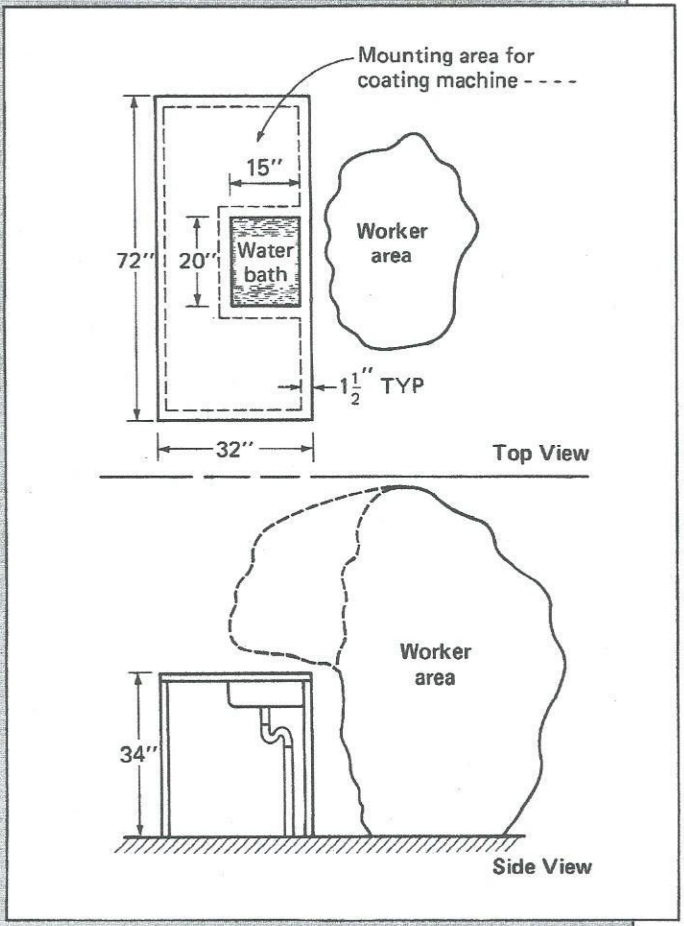


Fig. 7 (a) Aluminum plate.



(b) Dipping workstation.

Electrical Contacts

An electronics company is manufacturing a new portable computer. As part of the overall design, three batteries are needed to power a time clock. The company needs you to design the electrical contacts for holding these batteries and connecting them to a printed circuit board. The specific design requirements are:

- *Batteries*. Three type LR44, in series.
- *Envelope*. The components must fit within the plastic case walls. See Figure 8. The walls may have slots for contacts and locating features, etc. The envelope has five walls. The sixth side (where the batteries go in) must remain open. Another component, designed by someone else, will butt up against your envelope and keep the batteries from falling out. The dimensions given are for the interior of the envelope. Wall thickness and inside shape are up to you, but should not exceed 0.06-in.
- *Contacts*. The contact force shall be 0.1 lb. minimum and 1 lb. maximum at the PCB and at the batteries. The contact plating shall be nickel. The PCB contacting area has a diameter of 0.1-in. ±0.005-in. The contact locations are shown on the attached sheet. Any contact locations for *X1* and *X2* will be compatible on the PCB. (The design of this board has not been finalized.) The contacts cannot extend below the bottom of the envelope.
- *Assembly*. The computers will be assembled by robots, so the contacts will be handled and fit into place by a robotic end-effector (yet to be selected, but probably a 0.25-in. suction type). The envelope has an upper and lower half. During assembly, the lower half will already be in place. After the electrical contacts are in place (the part you are designing), the PCB and the top half of the envelope will be set in place.
- *Quantity*. 50,000 units will be assembled per month for three years.

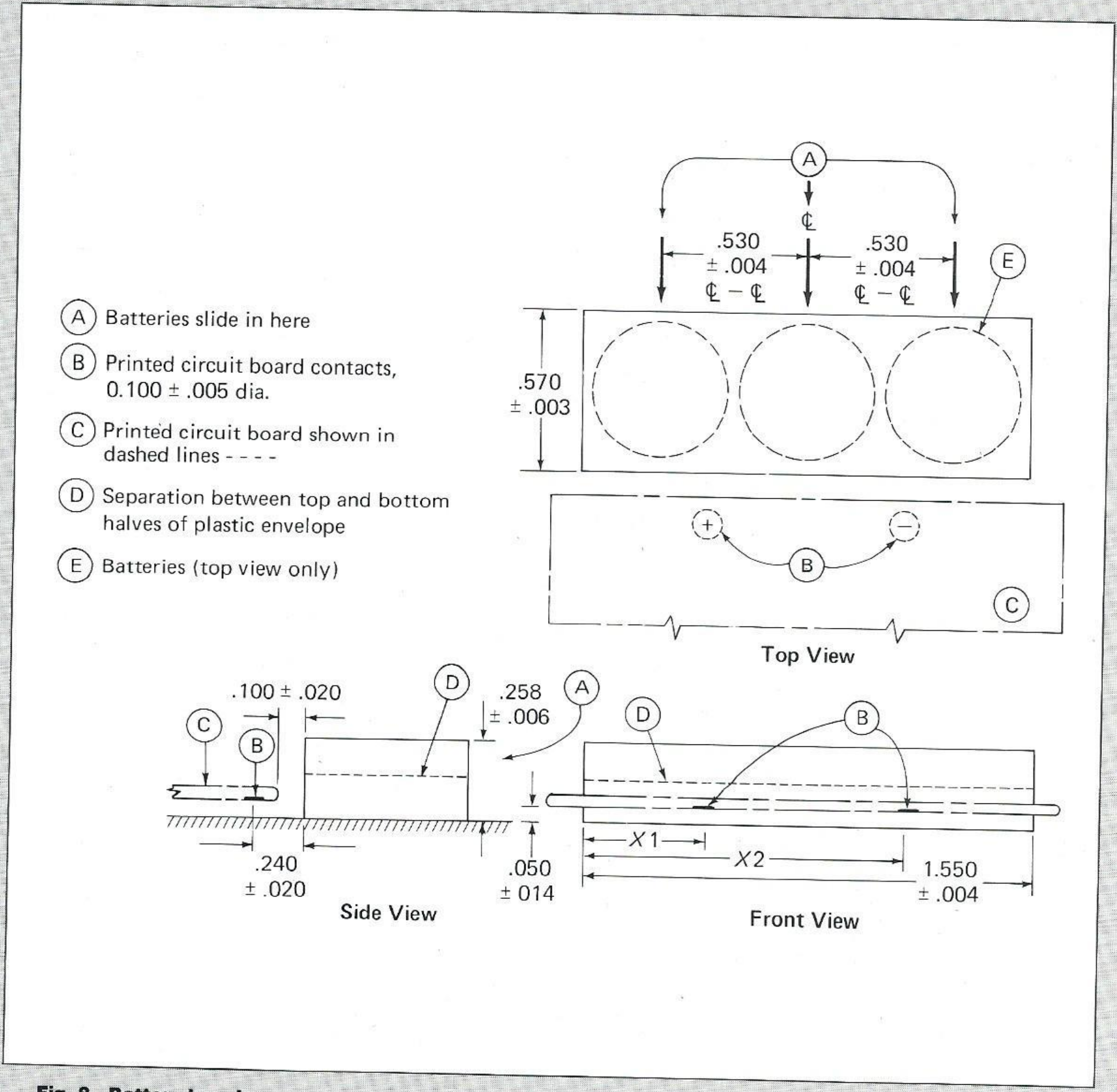


Fig. 8 Battery housing.

Protocol Analysis

Protocol analysis allows us to gather all the information a subject can verbalize *and* communicate through drawings and gestures. Its methods have been applied to problems in psychology and artificial intelligence, and the method itself has been extensively studied and tested to understand its limitations.

Other approaches to learning about the design process include retrospective and informed reporting. With retrospective reporting, a subject explains, at some later time, what he or she did. However, as studies have shown, subjects often do not report exactly what happened, but what they perceived happened. There can be a big difference between the two [1].

With informed reporting, an observer simply watches a person solve a problem and asks questions. Observational techniques, while they are less disruptive, can only record what is written down or retained in other tangible form. Ericsson and Simon summarize this research [1], and conclude that protocol analysis does not make the problem solving process artificial, and that it is the best known method for extracting a subject's thoughts.

But protocol analysis does have its drawbacks. One is that "thinking out loud" tends to slow subjects down. Nevertheless, research has shown that this does not affect the order or content of subjects' problem solving steps [1].

Another drawback of verbal protocols is that they cannot record what happens during incubation, when the subject "mulls over" a problem and generates ideas. Psychologists are divided over this issue in ordinary problem solving, and it is certainly a central question in creative design [2]. To control for this, we had the subjects summarize the state of their design at the end and beginning of each session, and held sessions on consecutive days.

A third problem is that protocol analysis is best suited to the study of subjects solving problems by themselves. However, engineers often work in teams, or at least pass their problems around to others for review. We could not think of any practical way of accommodating this aspect of design in our study.

Protocols. Each problem was designed to take about 10 hours to complete, which is very long compared to ordinary protocol analysis studies. The software design studies (see below), for example, were based on sessions lasting only two hours.

The length of time was dictated by a need to use realistic design problems, and by our desire to obtain data on all phases of the design process. Ideally, even longer protocols could be taken, but the 10-hour periods present formidable data analysis problems, and longer protocols would not have been analyzable.

Other Studies

Two large-scale studies have employed protocol analysis in learning about the design process in software engineering. One, by Adelson and Soloway [3], studied professional software engineers designing an electronic mail system, while the other, by Kant, Newell, and Steiner [4–6], studied graduate students designing algorithms for computational geometry.

Adelson and Soloway also studied the differences between expert and novice designers working on familiar and unfamiliar objects in familiar and unfamiliar domains. Similar results were found by both groups:

- *Designers rapidly developed a kernel idea and refined it during the design process.*
- *Designers kept a current mental model of the state of the design that was transformed, as the design progressed, from abstract to concrete.*
- *Designers made a great effort to keep the development of the design balanced.* They focused their efforts on parts of the design that were most abstract, attempting to keep all parts of the design at the same level of detail.
- *Designers spent about half their time simulating the behavior of their own programs.* The simulation process served many functions. It helped the designer integrate constituents from several parts of the design; it served as a kind of agenda to keep track of subtasks requiring attention; it encouraged a kind of balanced, methodical refinement of the software system; and it allowed for comparison to the design goal.

In software design, this goal is, in essence, a data flow behavior. In these studies, simulation means modeling the data flow through some part of the program at the current level of abstraction. This is the "working model" of the current design. Thus, simulation allows the behavior of the current state to be modeled so that a comparison with the goal can be made. Kant also reports that simulation helps the designer identify interesting opportunities for improving the design [3].

- *Designers took both mental and written notes on things to remember later in the design.* These included constraints, partial solutions, potential inconsistencies or other concepts that arose during the design process. These were not handled immediately, since they were at a greater level of detail than the current state of the design. Note-taking was not observed if the designer was working in a familiar domain but on an unfamiliar object. □

References

1 Ericsson, K. A., and Simon, H. A., "Verbal Reports as Data," *Psychological Review*, Vol. 87, No. 3, pp. 215–251.

2 Weisberg, R., 1986, *Creativity*, New York: W. H. Freedman and Co., 1986.

3 Adelson, B., and Soloway, E., *A Cognitive Model of Software Design*, Report Nr. 342, Dept. of Computer Science, Yale Univ., 1984.

4 Kant, E., "Understanding and Automating Algorithm Design," *Proceedings*, IJCAI, Los Altos, CA, 1985, Morgan-Kaufmann, pp. 1243–1253.

5 Kant, E., and Newell, A., "Naive Algorithm Design Techniques: A Case Study," *Proceedings*, European Conference on Artificial Intelligence, Orsay, France, July 1982, pp. 40–51.

6 Kant, E., and Steiner, D. M., "Symbolic Execution in Algorithm Design," *Proceedings*, IJCAI, 1985, Los Altos: Morgan-Kaufmann, pp. 225–231.

Methodology

We presented the problems to a total of six subjects, with three subjects for each problem. Of the subjects, four were experienced, practicing design engineers in areas similar to their assigned problem. Two experienced engineers were assigned to each problem. The other two subjects were graduate students who had some limited design experience. There was one graduate student assigned to each problem.

The distribution of subjects is shown in Table I. The results included in this paper are based on the partial reduction of data taken on subjects S2, S5, and S6.

Refining Problems

During a one-hour session, we had a graduate student (not one of our six subjects) read a problem and develop a rough conceptual design. Our purpose was to make sure that all essential information was available, and to identify any aspects of the written description or the initial sketches that might confuse the subject. We also wanted to see what kinds of questions the subjects might ask, so that the examiner could be better prepared for the subsequent data collection.

The second step was to take a full 10-hour protocol for each problem from a graduate student with limited industrial experience (subjects S1 and S4). The purpose was to collect data on less experienced subjects, and to make sure the problem statements were clear.

Collecting Data

Next, we administered the problems to experienced engineers who were working as designers. Each subject's background was matched to a problem as closely as possible.

Visual protocols. To record a mechanical designer's words alone would be to miss all of his or her gesturing, sketching, and pointing. Thus, the protocols were recorded on videotape to capture all data possible.

The camera was focused on the subject during the initial conceptual stages of the design when there was much gesturing and little drawing. Later, it was aimed directly on the engineer's sketch pad as problem solving became more detail-oriented. The camera was refocused as needed.

Physical setting. All sessions for the professional engineers (the "expert" subjects) were conducted at their places of employment. This was desirable because the subjects' reference materials were there, and because the sessions demanded a large commitment of subjects' time.

In a conference room at the plant, a videotape recorder and camera (plus a back-up audio recorder) were set up and tested prior to the subject's arrival; this was done to deemphasize the equipment's presence and to help subjects relax. Most of the subjects forgot the camera was there after working a while, as evidenced by their comments and by the examiner's perceptions. Both audio and video recordings were made as a backup.

Incubation. Interruptions, such as those between sessions, were not always avoidable. A subject could verbalize his or her thoughts for two to three hours, but after that, the subject lost concentration and became too tired to continue. Since the solution time for each problem was about 10 hours, the sessions were conducted over the course of two to four consecutive days.

At the conclusion of one session, the subject was encouraged not to think about the problem until the next. Despite subject's claims to the contrary, some thinking inevitably took place. To help account for this, subjects were asked to summarize their designs at the end and beginning of each session, and sessions were held on consecutive days.

Examiner. During the entire protocol session, the examiner was in the conference room with the subject, monitoring and operating equipment, changing tapes, and answering questions.

The subjects always had questions, and sometimes were unsure about the scope of the problem, but the examiner tried to keep them from delving into fringe areas. Because of the nature of the questions, it was important for the examin-

Table I
Distribution of Subjects

Problem	Grad Student	Professional
battery contacts	S1	S2 and S3
flipper dipper	S4	S5 and S6

er to have a technical background. (Here, the examiner was a mechanical engineering Ph.D. student with four years of industrial design experience.)

Another of the examiner's key tasks was to keep subjects verbalizing, since they often became lost in thought. When this happened, the examiner said only "keep talking, please," and nothing more.

Procedure

At the beginning of the first session, the examiner presented the subject with a set of written instructions to ensure consistency among subjects. The subject read them to him- or herself, with the examiner answering any questions about the verbalization process.

The first session continued with the presentation of a practice problem, which allowed the subject to become accustomed to thinking aloud, check out the equipment, and feel at ease. The examiner had subjects abandon their solutions when verbalization seemed routine. Subjects were encouraged on their performance, with suggestions made and questions answered. Then, the real problem solution began.

Subjects were given the written problem statement and asked to read it out loud. This was done to reinforce the verbalization needed in the remainder of the sessions. Subjects were encouraged to work at a natural pace.

We supplied subjects with pen and paper for their sketches. They often complained about using pen, but we didn't want to lose information that was erased. Pages were numbered so that we could keep track of them while analyzing the protocols. (It was essential to have copies of the sketches while watching the videotapes, since much of the protocol data is in the drawings.)

One subject even brought in drafting supplies and paper to construct a full size drawing of his machine. This behavior was encouraged, if it made problem solving in the experimental environment more similar to the subject's everyday work experience.

Use of CAD. Although two of the engineers occasionally used CAD systems, we did not allow their use. We did not know what effects CAD would have on the problem solving process, and we did not want to introduce another variable. (This is an area for further study.) ☐

Toward Expert CAD

Continued from page 65.

Conclusion

Our observations have important implications for the study of design methodology and the development of AI-based CAD tools. In design methodology, it is important to develop objective criteria for judging the quality of the design process as well as the quality of the final design. Many of the observed behaviors have been labeled as suboptimal by design theorists.

The protocol data suggest that such judgments may be premature. There may be very sound reasons why human designers work in this way, and further study is needed to determine whether such behaviors are in fact inferior, and if so, whether they result from poor training or from cognitive limitations.

In any case, the data also suggest that future AI-based CAD tools must be highly flexible, allowing the designer the freedom to depart from balanced development or to pursue opportunistic search strategies. In addition, future CAD tools have an important role in helping the designer manage the complexity of the design, for example, by keeping an agenda of outstanding problems and tasks, and by providing a smooth conceptual interface that helps him or her grasp the entire state of the design and the interactions among various components.

It is our hope that further analysis of our protocol data will help sharpen these observations and clarify many of the outstanding issues mentioned above. The development of an empirically-justified theory of design will provide a sound basis for the development of future CAD tools and for the training of future mechanical engineers.

Acknowledgements

The authors gratefully acknowledge the support of the National Science Foundation under grant number DMC-8514949.

References

1 Love, S. F., *Planning and Creating Successful Engineered Designs*, Advanced Professional Development, Inc., Los Angeles, CA, 1980.

2 Ostrofsky, B., *Design, Planning, and Development Methodology*, Englewood Cliffs, NJ: Prentice-Hall, 1977.

3 Hubka, V., *Principles of Engineering Design*, London: Butterworth Publishers, 1982.

4 Pahl, G., and Beitz, W., *Engineering Design*, London: Springer-Verlag, 1984.

5 Rabins, M., et al., "Design Theory and Methodology—A New Discipline," *Mechanical Engineering*, Vol. 108, No. 8, pp. 23–27.

6 Ericsson, K. A., and Simon, H. A., "Verbal Reports as Data," *Psychological Review*, Vol. 87, No. 3, pp. 215–251.

7 Newell, A., and Simon, H. A., *Human Problem Solving*, Englewood Cliffs, NJ: Prentice-Hall, 1972.

8 Mitchell, T., Steinberg, L., and Shulman, J., "VEXED: A Knowledge-Based VLSI Design Consultant," Rutgers AI/VLSI Project, Working Paper Nr. 17, 1984.

9 Adelson, B., and Soloway, E., *A Cognitive Model of Software Design*, Report Nr. 342, Dept. of Computer Science, Yale Univ., 1984.

10 Tong, C., "Knowledge-Based Circuit Design," doctoral dissertation, Dept. of Computer Science, Stanford University (publication forthcoming).