1 Relations between Cognitive Psychology and Computer System Design

Thomas K. Landauer

Cognitive psychology is more intimately related to the design of computers than to that of traditional machines, such as automobiles and home appliances. There are several reasons. First, the new information technology is so flexible that functions change with bewildering frequency. It is ever less feasible to count on the existence of experienced operators. Unlike typewriters and automobiles, it seems unlikely that information machines of the future will stay the same long enough for public school training to prepare people for lifelong careers based on their use. Thus easy learning or self-evident operations are critical. Second, and equally important, the tasks for which computers are the tools are generally ones in which the human's thought processes themselves are being aided. The maturation of computer applications is taking us ever farther in this direction. The first jobs for computers involved routine information tasks like bookkeeping, in which mechanical procedures once done by humans could simply be assumed by machines. Computers increasingly are used to support dynamic interactive tasks, like text editing and financial simulation, in which the user's mind is an important and lively component of the total system. Designing tools for this kind of activity is an intimately cognitive-psychological activity. Its accomplishment can no longer be viewed as that of first designing a machine to do something, then designing the controls by which the operator guides the machine.

Although the need for greater consideration of users has been recognized for some time now, the response so far, by and large, has been shallow. In attempting to provide greater "user friendliness," designers and programmers have indeed paid more attention to the usability of their systems, and in doing so have exploited the much expanded power of the systems with which they work. For example, they often use the larger memories now available to store larger programs that are supposed to better support usability. But this has been done without much basis other than individual designer intuition and common sense. Undeniably, common sense, combined with some vigorous trial and error, has already done much to improve systems. Just as undeniably, however, there is much farther to go than we have come. People still complain bitterly about the difficulty of learning to use even text editors and spread sheets, which are among the most thoroughly evolved interactive devices. And there are as yet only a few cognitive tools available that offer totally new ways of accomplishing mental tasks—symbolic math languages are one example—although one would think that the capabilities of computation would open the way for hosts.

Psychologists have become increasingly involved in the design of new computer systems and in research and theory aimed at understanding the human component of the problem. For example, before divestiture, Bell Laboratories alone employed around 200 psychologists in work related to computer system development. Many times that number are employed by other software and hardware companies the world over, so psychology apparently has something to sell. Indeed, talking to applied psychologists and managers in such settings usually elicits tales of success. But tales of dissatisfaction and frustration are also common. It would not be altogether unfair to characterize the situation as follows. Although psychologists have brought to development efforts a dedicated professional interest in user problems, and have successfully acted as intelligent advocates of the interest of users, they have not brought an impressive tool kit of design methods or principles, nor have they effectively brought to bear a relevant body of scientific knowledge or theory.

In addition to helping to build better systems, one would also hope that the interaction of cognitive psychology with design would help to advance the science of mind. Many computer systems are created to interact with, aid, or replace mental processes. Surely the problems encountered in trying to make them do so ought to feed new and interesting psychological research. Computer technology should provide better opportunity for applied research that can contribute to the science of mind than anything we have had in the past. It offers an arena in which potential understanding of human mental powers and limitations can be tested.

The iterative interplay between the invention of new methods to support cognitive activities and the analysis of their successes and failures is a very exciting prospect. In my view, cognitive psychology has suffered from the lack of an applied discipline in which the completeness of its accounts could be measured, or from which a sorting of phenomena into those important for actual human function from those of merely scholastic interest could be made. Physics and chemistry have had engineering, physiology and biology have had medicine to play this role. But cognitive psychology has had only limited associations with artifact invention, primarily because the most interesting aspects of human cognition are complex information processes for which, until recently, the means for constructing useful new tools were lacking. We now have the opportunity, but we have not done much with it yet.

What I am concerned with here is the development of a more fruitful interconnection between the science of cognitive psychology and the science, art, and engineering of computer systems. What follows is a brief analysis and survey of the ways in which cognitive psychology has and can interact with computer system design. It offers some examples of work already accomplished, but only as illustrations. No new substantive contribution to the field is reported here. The analysis I propose is quite simple and straightforward. I suggest four principle ways in which cognitive psychology can interact with computer system invention and design.

1. We may apply existing knowledge and theory directly to design problems.

2. We may apply our armamentarium of psychological ideas and theoretical machinery to the creation of new models, analyses, and engineering tools.

3. We may apply our well-developed methods of empirical research and data analysis to the evaluation of designs, design alternatives, and design principles.

4. We can use problems encountered in design to inform and guide our research into the fundamentals of mental life

Each of these relations is discussed in somewhat more detail, with, in each case, a few notable current issues being given particular attention.

1 Application of Existing Knowledge and Principles

One potential way to relate cognitive psychology to system design is to search for existing knowledge and theory that bears on recognized design problems. One problem that seemed ripe for this approach was the assignment of names to commands. Many interactive systems require the user to enter one or more characters as a cue for the system to perform one or another operation; these are known as commands. For example, s/hte/the might be the command needed to change "hte" to "the" in a line, where "s" is the abbreviated name of a "substitute" command. The character string "name" can be assigned at random as long as it can be a unique code for the system to interpret (the necessity of uniqueness can sometimes be relaxed if interactive disambiguation is feasible). In many applications people appeared to have difficulty in learning to enter the right strings to effect the correct operations, or at least that was one way to describe their struggles and confusions with new systems. Various people—users, critics and even programmers (see Norman, 1981)—blamed the difficulty of learning new systems on poor selection of command names. Because the learning of the names of commands appeared to be extremely similar to the classic laboratory paradigm of paired-associate memorization, psychologists interested in the matter thought they had found a perfect opportunity to apply things they knew. Paired-associate learning is a very well-developed area, with a rich literature of findings and phenomena and a goodly number of reasonably accurate, if somewhat restricted, theories.

The most directly applicable findings seemed to be a constellation of results showing that almost any kind of prior knowledge of the stimulus or response member of a pair in which the learner was to learn to associate the response to the stimulus, or any prior degree of association between the two, caused more rapid learning (as, for example, reviewed in Goss and Nadine, 1965, Postman, 1971, or Underwood and Schultz, 1960). To summarize roughly what the laboratory findings seem to say regarding the command name learning situation, they suggest that the response terms, the "names," should be "highly available." Availability means that (1) the names should be regularly spelled, well-known words that the user will not have to learn as words or as spellings, and that are common in the use of language to which they are accustomed, and thus easy for users to think of; (2) they should be words or obvious abbreviations of words that are related in meaning to the topic of learning, that is, to the category or meanings of other names in the set being learned; (3) yet they should be discriminable from other names in the set as much as is needed so that no two responses will be improperly confused with each other; and (4) if feasible, the words that are to be responses, that is, the names, should already have some "natural" association with the stimuli, that is, with the mental or environmental conditions in which they are appropriate.

Attempts to apply this set of principles to the selection of command names, and more especially the research aimed at demonstrating an advantage of so doing, are instructive in a number of ways. The earliest reported experiments (e.g., Black and Sebrechts, 1981) used a typical laboratory paper-and-pencil version of a paired-associate task, except that they substituted some representation of the nature of the computer operation for the stimulus and the name of the command for the response. They studied text editing operations that were described either in a phrase or by before-and-after examples of the effect of the commands. In some versions the "responses" were intuitively assigned names, those that some existing system uses. In other conditions, names were chosen by asking college students to suggest terms that they thought would be appropriate for the operations. As expected, superior learning occurred for paired-associate lists when presumably more discriminable and highly associated names were the responses.

At about the same time, Landauer, Galotti, and Hartwell (1983) performed an experiment in which they taught the first half-dozen commands of an actual text editor to typing students, with different kinds of names given to the commands for different groups. One set of names, "add," "omit," and "change," was chosen in an elaborate procedure of eliciting, from other typing students, the verbs that they would use to describe the operations of editing that those commands are designed to effect, in a setting in which they imagined themselves to be instructing another typist. For a second group the commands, "append," "delete," and "substitute" were the words chosen by the editor's original programmers, and in a third group the commands were named with randomly chosen unrelated words, "allege," "cipher," and "deliberate." There was a slight, but far from significant, advantage to either natural names or the programmer's names over the unrelated words, but no appreciable mean difference between the more natural and the programmer selected command names.

Barnard et al. (1982) also failed to demonstrate overall performance time effects for command name choice in an interactive text manipulation context, and Scapin (1981), in a paper-and-pencil study, found less common names easier to learn.

Subsequently, Grudin and Barnard (1984), determined, one might infer, to exonerate experimental psychology, conducted another experiment. Adult subjects learned 12 commands for a system that had many of the characteristics of a text editor, but was designed to have pairs of commands that were similar in function except for the objects to which they applied, e.g., whether the command put a word at the end of a sentence or at the beginning. In this somewhat simplified and artificial interactive setting there was a significant advantage for experimenter-chosen "specific" over unrelated words.

I do not propose to resolve the apparent conflicts between these various experiments here (but see Landauer and Galotti, 1984). Rather, I want to use them to illustrate two simple points. First, it is possible for some very well-established phenomena that have robust effects in abstracted laboratory tasks to have very different levels of influence when embedded in more complex tasks. The system taught by Landauer et al. was difficult to learn; beginners made many mistakes in applying command names and were confused and frustrated. Indeed they manifested just those difficulties that had led many critics (Ledgard et al., 1980; Black and Sebrechts, 1981; Norman, 1981; Carroll, 1982) to propose the use of more "natural" names for commands. Nevertheless, simply using "natural" names had no appreciable benefit, and much more extreme and contrived contrasts proved necessary to demonstrate naming effects. Apparently something else makes the learning of text editors difficult. Even though it is demonstrably easier to learn names with appropriate prior characteristics (e.g., Black and Sebrechts, 1981; Grudin and Barnard, 1984) in special circumstances, such factors do not appear to contribute much to the overall difficulty of learning the rudiments of interactive computing under ordinary conditions.

The point of applied research is to understand what matters in realistic contexts. In regard to naming, then, the proper conclusion from the work so far is not that an effect is there if you look for it in the right way, although this is true. The important lesson is that what really makes things difficult must be looked for elsewhere. A second point of interest also derives from these experiments. A different variable concerning the assignment of command names, the manner in which differences between names were "mapped" to operations requiring syntactic constructions following the name, appeared to make a very large difference in ease of learning. For example, in some versions of the experimental editor, removing a word within a line required the command construction "name /hte//", but removing a whole line did not require the slashes or the input of the incorrect text. Using the same command name for both, whether "substitute" or "delete," created marked confusion relative to using substitute for one case and delete for the other. This was equally true if the two names were the unrelated terms "cipher" and "deliberate." Nothing in the traditional verbal-learning literature corresponds well to these mapping differences, and previous psychological knowledge therefore affords little help (or hindrance.)

Somewhat more success has been achieved in a few other attempts to apply prior knowledge. (However, it must be admitted that for both of the examples to follow, insufficient replication and critical notice has yet been reported to justify much bragging.) One example is a recent attempt by

Landauer and Nachbar (1985) to apply some well-known laws governing decision and motor selection time to menu choice. In a typical menu-driven access to a large database, users are given a series of choice frames by use of which they guide an hierarchical tree search. For example, the first frame of a dictionary search interface might present the choices (1) apple—mother (2) motor—trouble (3) under—zebra. If the user chooses (1), the next screen would present a further subdivision into (1) apple—cap (2) car—keep (3) key—mother, and so forth.

The problem is this: Given that items are selected by indicating one of *b* alternatives in each of several successive hierarchically organized menus, and given that the total set can be meaningfully subdivided in many different ways, what menu structure is optimal? Two laws are potentially relevant.

1. Broader menus necessarily require a decision among more alternatives. Hick's law (Welford, 1980) states that mean response time in simple decision tasks is a linear function of the transmitted information, which, for equally likely alternatives, gives

$$rt = k' + c' \log_2 b, \tag{1}$$

where *b* is the number of alternatives, and k' and c' are constants.

2. Sometimes more alternatives per screen will require targets that are physically smaller and harder to select. Fitts's law (1954) states that mean movement time is a function of the log of distance (*d*) over width (*w*) of target: that is,

$$mt = k + c \log_2 (d/w). \tag{2}$$

In a touch-screen menu, for example, the user touches an appropriately indicated area on the screen, the target, to indicate a choice. The target's distance is nearly independent of the number of alternatives, but the width may decrease as the screen is divided into more regions. In Landauer and Nachbar's experiments the available screen height was equally divided among the choices, so the touch target width varied inversely with the number of alternatives. For this case, then,

$$mt = k'' + c'' \log_2 b.$$

If the decision and movement components are assumed to be accomplished in succession, their mean times will add, giving mean choice time, *ct*, for a menu of *b* items

$$\begin{aligned} ct &= dt + mt \\ &= k' + c' \log_2 b + k'' + c'' \log_2 b \qquad (3) \\ &= k + c \log_2 b. \end{aligned}$$

Data from experiments with 2–16 alphabetically and numerically arranged alternatives fit the predicted functions closely. This particular functional form has an important implication for the design of hierarchical menu schemes. The total search time from root to leaf in a symmetrical search tree is given by

$$\begin{aligned} T &= \text{(number of steps)(mean time per step)} \\ &= \log_b N \text{ (time per step)}, \end{aligned}$$

where b is the branching factor, i.e., the number of alternatives at each step, and N is the number of terminal nodes. Substituting the mean time per step for human choice as given in equation (3) yields

$$T = \log_b N(k + c \log_2 b).$$

Note that the additive constant k now should include the system response time to change menus. On multiplying out,

$$\begin{aligned} T &= k \log_b N + c \log_b N \log_2 b \\ &= k \log_b N + c \log_2 N, \end{aligned}$$

we see that b, the number of alternatives, affects the total time only by determining the number of steps. Therefore, for situations in which equation (2) holds, search time will be minimized by using as broad menus as feasible within space and display constraints.

The empirical menu-use experiments confirmed this expectation; very narrow and deep menu sequences took as much as twice as long to traverse as very broad and shallow ones. The question of how far this analysis generalizes needs more research. In a second, unpublished, experiment, Nachbar and I used key-controlled cursor movement instead of pointing for target selection, and 4–64 alternatives, and got equally good fits. But the expected results for choice domains less well ordered than alphabetically or numerically arranged sets are still an open question. Variations in depth and breadth of menus for computer concepts and other semantic categories have produced less clear results (Miller, 1981; Snowberry, Parkinson, and Sisson, 1983; Tullis, 1985). One aspect of such studies is that the division of a natural set into more or fewer arbitrary partitions does not necessarily yield equally good categories for human recognition, or ones to which equally comprehensible names can be assigned. In other words, the "naturalness" of the categories needs to be considered, as well as the information processing time functions for equally good alternatives as described by equation (3).

Another unresolved issue is the effect of errors. In the cursor-selection experiment, total errors, as well as time, decreased with greater menu breadth, as they did in Tullis's (1985) study of semantic categories. However, in using touch screens with many alternatives, thus very small targets, total incorrect responses increased. Errors are likely to be more costly, on average, for deeper menus where larger error recovery paths are possible. Nevertheless, further studies are needed to define the effects for various modes of error correcting.

A final issue raised by these experiments was the relation between performance and preference. On questioning, participants did not like best the conditions that led to the fastest and most accurate searches. The relation between these outcomes has not frequently been considered in human information processing research, but is an important practical concern and poses an intriguing theoretical problem for cognitive psychology broadly construed.

Another example of applying principles is Card's application of Fitts's law to the comparison of pointing devices. (Card, English, and Burr, 1978; Card, Moran, and Newell, 1983). In this instance, the principle was used as a way of describing performance on a set of devices, mouse, joystick, and keys, that had been invented without recourse to any formal psychological knowledge. The result was an elegant and revealing analysis of reasons for the superiority of the mouse over the other devices in the task studied and the invention of a quick engineering test for proposed pointing methods. Card used the good fit to Fitts's law to argue that no pointing device would be likely to do much better than the mouse, since its constants, i.e., information transfer rates, were about as low as any previously observed in motor-control experiments. This is a valid and extremely useful conclusion, at least if properly restricted to the tasks on which it was based, and to the relatively passive devices so far developed for the purpose. What I wish, however, is that such analyses, instead of leading only to intelligent choice among existing alternatives, would be used to point the way to the invention of new methods. Card et al. show that a fundamental limit to the performance with today's pointers is the eye-hand control rates of the human operator. What they did not undertake was the application of their understanding of human pointing to the specification of a device that would improve the human performance, an aid rather than a mere transducer. Perhaps some of the human system's noise could be overcome, or some intelligence could be applied to infer the desired target more accurately. As I shall remark below, such challenges are of special interest to the application of psychology in this area.

I am at a loss to come up with any better examples of the application of prior knowledge or theory in this field, or even many more to equal them. I find this dismaying. Articles and books claiming to give guidance for designers do list a number of basic facts about perception, memory, and problem solving that are believed to be important to designers. Some, such as specifications for the size, contrast, and font of characters to be presented on a CRT screen or colors and physical properties of phosphors and flicker rates are well founded and of obvious relevance to some design tasks. But they do not thrill one as applications of cognitive psychology. The more cognitive variables discussed usually fail to make contact with the designer's job in an easily understood way, or at least in a way that has given rise to empirical demonstrations of value. For example, one of the most frequently cited principles is the fact that short-term memory can only be relied on to maintain around five chunks. Setting aside the practical difficulty of identifying "chunk," I doubt that knowledge of this principle has often influenced a design. I suppose a perverse designer or a diabolical experimenter could figure out a way to make an interface require working memory for 20 items, but I have yet to see this principle actually put to use in a convincing manner.

There may be several reasons for the lack of successful application of known principles. One is that not enough people really understand the principles or their basis. One example is a recent theoretical paper on the menu breadth problem discussed above. It assumed choice time linear with number of alternatives on the basis of Sternberg's findings for memory set scanning with a fixed number of alternative responses, a vaguely analogous but actually irrelevant principle. Probably there are too few people who understand both psychological principles and systems with sufficient depth. Perhaps this will change as more sophistication on both sides develops.

Another and more fundamental reason for limited application of principles is that the way in which cognition has been studied in the psychology laboratory has often promoted an interest in variables of only theoretical importance, ones that can reveal something about the workings of the mind—but not necessarily ones having large and robust effects. The factors that actually account for the difficulties encountered in real life are not necessarily objects of study. A great deal of experimental research in cognitive psychology has been concerned with hypothesis testing, where the existence of a qualitative result—that something has or has not a given effect—may decide a theoretical issue. This is quite a good strategy for advancing theory because it finesses a large number of difficult questions of measurement, scaling, and explicit modeling assumptions. However, it carries with it a greatly diminished interest in the true size of effects, so much so that it has become depressingly common for journal articles to report significance levels without giving data on the differences between means or distributions. Authors adopt a language in which there are said to be effects or no effects, as if the world exists in binary states. Although a great deal can be done to build theories and understand nature with this kind of black-and-white picture, it tends to leave us in a rather helpless position when it comes to choosing principles on which to found technology or understand peoples' behavior in its natural setting.

Another reason for the applicative poverty of cognitive psychology is that the theories we have pursued have led away from rather than toward attempts to describe in full the performance in any given task situation. Thus we have been, quite rightly, fascinated with analyzing the act of reading an isolated word so as to discover the processes and organization by which it is accomplished. This leads us into intricate and clever debates to resolve such questions as whether there is any stage at which individual letters are identified as such. A good deal of progress has been made in learning how to study and understand such hidden processes. However, while analysis has proceeded apace, the analog of chemical synthesis has been almost completely lacking. That is, we have not found out how much of what kind of process ability acquired in what way needs to be combined how with others in order to produce effective reading, or even word recognition. We have hardly begun looking for what factors have large effects and what small when in combination with what others. As a result, the intense and sophisticated work in this area has not yet contributed substantially to writing, calligraphy, typography, the teaching of reading, or the design of improved text presentation via computers.

Lest this sound too pessimistic, my view is that we have indeed learned a great deal about processes that will someday be of importance in the practical world, and perhaps especially in the potential world of computer aids. But we have not learned enough of the right things to make the knowledge productive for application. There are far too many holes, and too many uncertainties in what principles are relevant to what tasks. Nonetheless, it seems likely that we shall find the knowledge at hand useful as some of the building blocks we shall need. The fact that current knowledge is not sufficient on its own does not mean that it is useless—only that it will need supplementing and filling out to form the basis of a synthetic discipline.

2 Application of Theoretical Machinery

Cognitive psychology has developed a number of tools for thinking about mental processes that may be used to understand the tasks for which computer systems are designed, without there being any direct sharing of content. One example is the form of theory in which a complex human performance is modeled by a flow diagram or computer program that defines some set of component processes and the order in which they are accomplished. Starting with a view of skilled performance based on earlier cognitive theories (Miller, Galanter, and Pribraun, 1960; Newell and Simon, 1972) and deriving hypothesized subprocesses from observations and protocols of expert performance, Card, Moran, and Newell (1983) formulated on "engineering model" for predicting the temporal characteristics of interacting with a keyboard-driven system. The model makes little use of substantive principles from earlier discoveries in cognitive psychology. However, it has a very familiar form and style. For the stable performance of highly practiced operators of text editors, Card, Moran, and Newell obtained reasonably stable estimates of response time for subcomponents like the selection and execution of certain commands. Combining these according to the time-additive dictates of their flow model yielded predictions for total task performance times that were fairly good, i.e., within about 30% of observed times, for a certain range of tests.

It is less clear that the model has strong prescriptive properties, that it can tell the designer much about how to design a good system in the first place. For one thing, it can be argued that the model, as so far developed, underplays many of the important determinants of real performance with interesting systems—for example, aspects of learning, transfer, error generation, and nonoptimal selection of alternative methods by users (Olson and Nilsen, 1985). The model was developed to give predictions of the time that experts require for certain components of an overall system, given a detailed specification of the system design and some preliminary measurements. This is quite useful for comparing two or more versions that one might be considering, and for tuning a system under development. It is obviously a very significant step in an important direction.

It is certainly too much to expect all the problems to have been solved in one project. However, some of my goals for the field are exemplified by what is missing in this work. The engineering model approach seems aimed primarily at providing feedback evaluation for the design process, rather than fundamental knowledge useful as its foundation. Take the treatment of errors as an example. The main development in the Card, Moran, and Newell book deals with error-free performance. Errors are considered, but not what causes them or where to expect them, and only as another subcomponent of performance whose time costs can be estimated and modeled. That is, given a determination of how many errors of what kind will occur, the model can be extended to predict the time taken up by commission and correction of errors, as well as performance of correct sequences. But the data and theory are not brought to bear on what design features lead to what kinds of errors. In Card, Moran, and Newell's study of text editing, for example, my calculations from the published data show 35% of expert's time and over 80% of the variance in time to be attributable to errors.[1] This implies that analysis of the source of errors might lead to the design of much more effective editing aids.

The attitude seemingly implicit in the engineering model approach, and, I believe, too widely held in the human-factors world, is that invention and design are engineering acts in which scientific psychology has no part to play. The psychological analysis is used only to create a shortcut measure of usability, not to produce understanding and insight to guide the creative process. This is too modest, and dull, an aspiration.

A related application of a similar mode of theory is that of Polson and Kieras (1985). They used the machinery of a production system-based cognitive simulation program to model what a user must know in order to operate a system. They go on to assert that the difficulty of learning an operation will depend on the number of new productions needed to acquire the skill. From a scientific point of view there is something slightly circular in the method, because the choice of what and how many productions to use for a given of task is not rigorously specifiable. Nevertheless, in their hands, and potentially in that of people experienced in using their theoretical model, it is possible to make better than gross intuitive predictions about the relative difficulty of systems and of the amount of transfer between the learning of one and another.

Again the application of theoretical technique from cognitive psychology seems to have yielded the beginnings of an effective engineering tool. As in the case of Card, Moran, and Newell's keystroke models, what we might call the "mentalstroke" model offers more promise of description than of prescription. Eventually we would like cognitive psychological theory to tell programmers how to invent and design, but for the time being we should be delighted to have some methods that, even at a rather approximate level of precision, and even if they require a component of human judgment, are capable of helping us to evaluate one design versus another.

Another approach was illustrated first by some early—for this field—work of Phyllis Reisner (1981). In this the theory schema is borrowed primarily from linguistics rather than cognitive psychology as such. The idea is that action sequences in some kinds of interactive interfaces can be described by a grammatical structure in a manner similar to the description of linguistic utterances. Such descriptions can both expose inconsistencies in existing designs and offer a plausible basis for designing consistent rule-governed interactive methods. Her work, and others that have followed (e.g., Payne and Green, 1983) make is appear likely that systems susceptible to tidy grammatical descriptions will be easier to learn and use than those that are not. This seems a promising way to go, especially since one can imagine using grammars to *generate* important aspects of design. However, what grammatical descriptions will prove to be best in the sense that they give most leverage on the specification of humanly usable systems is still a very open question.

3 Application of Investigative and Analytic Methods

There are two very elementary but fundamental methodological facts that are taken for granted by all experimental psychologists, but astonishingly often fail to be appreciated by others. The first is that behavior is always quite variable between people and between occasions. The second is that it is feasible to obtain objective data on behavior. In system development environments, ignorance of these two truths all too often leads to an evaluation of new ideas or designs by the opinions of a handful of unrepresentative people. Psychologists with proper organizational support can make an extremely valuable contributions simply by insisting on observing sufficient numbers of representative users on a set of representative benchmark tasks, and taking some systematic measures of task time, errors, opinions, examples of error types, and both user and observer intuitions about sources of ease and difficulty. Designers are constantly surprised at the value of such observations. The reason they are so surprised is that the performance and reactions of real users are so various and so often different from their own (because a designer is also one of the set of variable reactors). What might be termed the "naive intuition fallacy," that every-one else will behave and react similarly to oneself, appears to be very widely and strongly endorsed. The only professionals whose training tends to disabuse them of this error are experimental psychologists who have the experience of running experiments with live human subjects and being repeatedly surprised by their behavior.

On the other hand, the training of most psychologists has featured hypothesis testing by factorial laboratory experiments in which the goal is to find some critical variable that will have an effect that confirms or disconfirms a theory. The application of psychological research in support of design is quite different. It often requires new ways of thinking and considerable ingenuity on the part of the scientist. For example, it is often desirable to evaluate the usability of a tool that does not yet exist and is going to be hard to build. Thus, Gould, Conti, and Hovanyecz (1983) found out how well a nonexistent speech recognition-based typewriter would satisfy by substituting a human for the missing electronic components. Other research aims at discovering the main variables or factors that will make performance easy or difficult. To do this requires a method that will reveal large mean effect sizes under realistic conditions.

To put this somewhat differently, what application needs most, at least at present and probably into the future as well, are exploratory research paradigms rather than hypothesis testing ones. Experimental psychology has not been particularly productive in evolving such paradigms. Nonetheless, many of the basic tools of our trade can be applied with considerable leverage. Here are a few examples. Several groups of investigators have studied the question of how abbreviations for things like command names should be formed (e.g., Ehrenreich and Porcu, 1982; Hirsch-Pasek, Nudelman and Schneider, 1982; Streeter, Ackroff, and Taylor, 1983; see Ehrenreich, 1985, for a review). The Streeter, Ackroff, and Taylor approach is a nice example of the exploratory paradigm. They first had representative users nominate abbreviations for a set of words like those that might be used as commands in a real system. They characterized and classified the kinds of abbreviations that people gave and discovered (as usual) that there was a lot of variability both between people and within a given individual in the apparent rules being used. Puzzled and prompted by these results, they went on to evaluate a number of possible schemes for assigning abbreviations, including (1) using for each command name the most popular abbreviation given by the subjects, (2) using a rule that captured the maximum number of abbreviations, and (3) using some other rule. What they found is that using any consistent rule for the whole word set is what is most important. Even abbreviations for words that would be spontaneously abbreviated otherwise by most people were learned more easily if they conformed to a common rule. (As a substantive matter it is worth noting that here naturalness—interpreted as the popularity of an individual abbreviation—was pitted against rule governedness and lost.)

A somewhat more extensive example of exploratory research in which variability in behavior figures prominently is the work on indexing by Furnas et al. (1983). In a variety of domains, ranging from editing command names to recipe databases, they studied the names users spontaneously chose for desired items. The data uniformly falsified the common intuition that there is one best or only a few good names for most objects. Instead there are almost always many names, no one of which is very dominant. Simulation models based on these data showed that giving every entry as many names as people spontaneously wished to look them up under, typically around 30, would increase first-try success from 15–20% to 75–85%, without unacceptable increases in ambiguity. Next, controlled experiments (Gomez and Lochbaum, 1984), in which a laboratory version of a recipe database was searched by homemakers, demonstrated that an actual interactive system with "rich aliasing" could achieve the predicted benefits.

Other research and analytic techniques have also been put to good use. For example, multidimensional scaling was employed by Tullis (1985) to produce an apparently very successful organization of a database for menu presentation and choice. Finally, Egan and Gomez (1985) have exploited a set of standard measurements of individual differences in cognitive abilities to analyze the difficulties posed to users by a text editor. In this case, exploration of the characteristics of the task was accomplished by exploring the characteristics of people who find it easy or difficult. They found, for example, that the older the learner, the harder it was to master text editing, and that most of the difficulty correlated with age involved the construction of abstract text locating commands. This in turn led to the discovery that so-called full screen editors are easier to use, and their difficulty much less dependent on the user's age, primarily because they allow a simpler mode of indicating the text to be modified.

There are, of course, many other examples. Only these few are mentioned to give a sense of the range of possibilities. The overall point is that with some ingenuity many of the classical techniques of cognitive psychological research can be turned into exploratory and analytic methods appropriate to investigating issues relevant to design. It may be worth repeating that, in this mode, little substantive psychological theory or knowledge is applied. Instead, and this is important, new phenomena are added to the purview of the field.

Before leaving this section, I would like to mention some technical difficulties in employing cognitive psychological methods in applied settings. Perhaps the most challenging problem is to achieve the proper degree of representativeness in the experimental situations chosen for study. We wish to understand real users performing real tasks. But how can we choose full-scale settings in a way that ensures generality? If we test a principle as it is embedded in one system—say, one powerful editor—what can we say with confidence about its effect in others? Usually very little. How shall we ever rise above this difficulty? The ideal way would be to study a sample of systems drawn representatively from all systems that exist or might exist. Clearly this is impossible, if only because there is no way to describe the universe from which to draw the sample. Another approach, represented in the work by Furnas et al. described above, is simply to study several tasks, but to choose them so that they differ from each other widely and in most of the ways that one can imagine would be relevant to the phenomena in question. For example, Furnas et al. studied spontaneous name selection for text editing operations, system commands, cooking recipe index terms, common knowledge lookup keys, and want ad categories. If they all yield results with the same implications, as in this case they all envinced great diversity, then a sort of inforrnal Bayesian inference allows one to conclude that such results are to be expected in most other situations as well. (Prior: a phenomenon not due to fundamental and general causes has high probability of varying greatly over cases. Observation: phenomenon observed in many disparate situations. Conclusion: causes probably general.)

Discussing robustness and generality brings up the issue of the use of statistics. The proper use of statistics in the kind of research being described here is somewhat different from its use in theory testing. When the issue is whether there is or is not an effect of a particular kind, then a significance test with a conservative acceptance probability level is appropriate. But when we want to know about the importance of variables or factors, we really are interested primarily in estimates of mean effect size. We still want to take account of the fact that behavior is intrinsically variable and not fool ourselves into believing that a particular effect size is a true value when chance may have thrown it off. For this we want to put confidence intervals around effects sizes. Psychologists are too sophisticated at significance tests and too inexperienced with confidence intervals. This point is often made in textbooks and discussions of academic research, but is frequently ignored in practice. It is of much greater import for applied research.

Here is another aspect of statistics that needs more careful consideration. Consider the problem of comparing two features or two systems. If one obtains some benchmark behavioral measurement on the two, what sort of statistics are wanted? Suppose the decision is simply which is best, as in "Which font should I use?" Then a statistical test is irrelevant. One should just choose the one with the best mean effect. Now, obviously, if only a small number of subjects had been run and the data are highly variable, that judgment may be little better than a guess. But the only solution is to get more or better data if one wants higher confidence; a statistical test will not help a bit. On the other hand, putting confidence intervals on the data will help one know how good the bet is.

Put this another way. You are choosing between incorporating feature *a* and feature *b* and you have tested the system with both. You do a standard statistical test and get a significance level of .33 favoring *b*. The way to read this is that chances are 2:1 that (*b*) is at least a little better than (*a*). Clearly, if all other things were equal about the choice, you would choose (*b*) even though a .33 significance level is often described in the academic psychological literature as "absolutely no difference." In the applications field we should get used to using statistical methods to arrive at posterior odds ratios of this kind and using them in our decisions. If one has compared two systems and wants to tell others which is better, mean differences and confidence intervals, along with an odds ratio, would properly communicate the needed information from the study.

Of course, the choice between systems or between features is seldom one in which everything else is equal; usually there are differential costs to the various possible choices. The machinery for making wise decisions under combinations of uncertainties and costs is available, but this is not something into which we can delve here. Suffice it to say that neither a yes/no significance test alone nor a mean difference alone is useful. If one wants to make an informed decision, the information needed is the estimated mean effect size along with an odds ratio and/or confidence interval.

I shall mention one last matter. It is sometimes asserted that psychological research is not useful for system design because it is too unwieldy, expensive, and slow. Usually people making this statement have in mind an elaborate experiment in which all of the potential features in a system at each of their potentially interesting values or instantiations are compared. It would be rash to claim that such experiments would not be useful, but certainly most of the criticisms of them are true. What is really wrong with the assertion of inutility is its implicit assumption that this is the only way to do psychological research on system design. In fact, at this stage of maturity in the field, it may be among the worst ways to proceed. The main reason is that the choice of variables, factors, and levels is bound to be very poorly motivated. They would have to be chosen intuitively by designers or from the infinite number of possible values available, clearly a matter of endless thrashing in a muddy morass. We have seen, above, that there are many firmer research paths to follow.

4 Applied Research on Computer System Design as a Source of Problems and Theory for Cognitive Psychology

The human mind is an artifact of human culture. Although it is not constructed by the deliberate design of a team of people, nevertheless it is just one realization of an infinitely pliable system, programmed by culture, education, the knowledge base of the society, and the demands of the tasks and environments in which it finds itself. What phenomena of human cognitive processes to study is a very difficult scientific issue. Granted that the mind is an extraordinarily flexible, general-purpose device that might perform a vast variety of tasks each in a vast variety of ways, which does the scientist who wishes to study the "real" mind choose?

I believe that we want to study the human cognitive system doing those things that it now usually does or those things that it will be doing in the future, or just possibly those that it used to do in the past. There is no sense in which we can study cognition meaningfully divorced from the task contexts in which it finds itself in the world. I assert that the human mind is a construction on the capabilities of the brain and the culture in which it is embedded, and that in studying it we must be sure that we learn about the way it would behave in just those contexts. Otherwise we shall be doing merely scholastic play (although I do, of course, admit that scholastic play may sharpen our wits for more serious work). To draw the example almost to an absurd length, consider the human cognitive system to be a structure like a bridge, and therefore cognitive psychologists to be those who are interested in understanding the structure and function of bridges. We may, of course, take the bridge apart and use the girders to construct an oil derrick or box in its bottom and float it in the ocean as a barge. But we shall then be unlikely to gain direct insights into bridges. We might learn something relevant, but we would have to be very lucky to do so. The science of mind is, I assert, not the science of what the mind can do in a laboratory situation artificially rigged to make it relevant to one of our theories, but what it does in a situation naturally or artificially rigged by itself and its culture. (I hasten to repeat, the abstracted laboratory experiment is an essential tool, but it must answer questions raised by nature, and its answers must be tested against nature.)

Given this view, the only way to find out what the mind is like, and to stumble across questions that need to be understood more fully, is to study it in its natural habitat. Since the mind does not have any set natural habitat, we need to study it in the habitats in which it frequently finds or wishes to find itself. At the moment it seems to wish to find itself in constant interaction with computer systems, so this is where we must track it down.

An added advantage of choosing computer use as a "real" task for analysis is that it also offers attractive avenues for synthesis. Once having identified the problems and issues involved in this aspect of human mental life, having then, perhaps, taken them to the laboratory and understood them better, we should be able to test the sufficiency of our knowledge by trying to design computer systems that will function better. Computers have obvious advantages for trying to construct tools based on knowledge of cognitive functions because they have the information processing power to become involved in interactive tasks that include communications, information storage, inference, problem solving, and the like. The opportunity to try to build artifacts that interact with human cognition was extremely restricted before the advent of computer systems, but is much more open now. A concrete example might help to clarify this point. If we really understood the fundamentals of reading and its dominating, or rate-limiting factors, we might be able to build ways of presenting text that would materially increase the speed and accuracy of comprehension. The flexibility of computer systems ought to help us realize ideas of this kind and test them, thus giving us a rather new and powerful way to try out our theories.

One of the chief advantages of using applied cognitive tasks as a source of research problems is that the level of analysis, the kinds of components and processes, that become part of our theories will be dictated by what is important to human cognitive life, rather than by pretheoretical preferences for some style of explanation on the part of the scientist. An example is found in the work on indexing and information retrieval mentioned above. This started with no preconceptions of what kinds of theory or psychology might be relevant, and was forced to consider the distribution of word use in reference, and thus such matters as the difference between the distribution of word use within people and in populations, and so forth. This level of analysis was unlikely to have arisen from traditional cognitive psychology, but turns out to be the one needed to understand a very significant problem in human cognitive life. If one reacts to the work that resulted as not being relevant to cognitive theory, I think one must think very carefully about which is at fault. I believe it is just this kind of confrontation that is among the most worthwhile outcomes of the developing relationship between cognitive psychology and computer system design.

5 Summary

The dominant characteristic of work on human cognition in the context of computer design is that computer systems and the tasks and problems with which they help people are large and complex. The tasks tend to involve many detailed cognitive processes going on simultaneously and successively, and the designs of interactive procedures have an enormous number of details, any one of which may affect the interactive performance. This causes many problems. It means that previous research findings and theories are likely, at best, to be relevant to only a small portion of the total situation. It means that testing the effectiveness of one total system against that of others is likely to yield information of extremely limited generality. What one will have tested is "one mess against another." So many differences will exist between one system and another that the features or factors to blame or credit will be obscure. By the same token, some traditional methods, like factorial experiments, will not be very practical. There are too many features to examine, and the likelihood of interactions among effects not included in the experiments is too great.

It seems to me that mutually beneficial relations between cognitive psychology and system design can nevertheless occur. The essence of the investigative procedure I advocate is as follows. The investigator, applied psychologist, or system designer should set up a situation in which people doing a task that is, one hopes, to be aided by a computer system can be observed and recorded in some detail. From this the range of problems that are encountered and the things that people, novice or expert, can do well are noted. In the system design process, often the difficulties noticed will have obvious solutions and a fix can be made in the next iteration. This process of tuning a design by iterative user testing has been found extremely effective, for example, in the development of the Apple Lisa and Macintosh systems, which profited from very frequent user tests conducted by the applications software manager himself. Notice that features found at fault in such a process are identified in a context that is very realistic and includes most of the complexity under which the user will actually operate. Note also that traditional experimental design is neither terribly useful nor necessary for this process. One wants to observe and discover the kinds of difficulties being experienced; guessing ahead of time what factors to vary and control in an experiment might be hopeless. Moreover, the appropriate statistical test is "Which is the best bet?" rather than "Are we sure?" so while more data makes one surer, large effects observed with a handful of subjects are a sufficient screening method to help improve tuning. It really does not matter a great deal whether sometimes the tuning is done wrong because the observation was really a chance result, at least if the decision is not costly in programming time or system resources. In any event, the question of how much data should be collected and what level of certainty achieved can be weighed against anticipated costs and benefits.

This is fine for system development, but what it does it have to do with the science of human cognition? I maintain that the same process is well suited for generating the problems and serving as a testing ground for investigations of true scientific value in cognitive psychology. When particular problems are encountered repeatedly and found to be severe, in the context of a fully working (or partly working) system, but the fix or cure for the problems is not obvious to the designer, one has identified a prime candidate for deeper research. These problems suggest variables that are important to human performance and that need studying. Moreover, having subjected them to whatever study methods are needed for fuller understanding, the psychologist is in a position to test that new understanding by trying to design a system that overcomes the problem. And again the test will occur in just the right context, embedded in all of the complexity in which humans function.

Thus, it seems to me, a research cycle that starts and ends—or at least starts and periodically return to—full-scale system design problems is a very promising way to do and use cognitive psychology. By no means is it the only good way. Certainly the more traditional and academic pursuits of trying to understand obviously important cognitive processes like perception and reading will continue to be the mainstream of the field. Nevertheless I believe that the increasing interaction of cognitive psychology with the particular design problems of a cognitively rich and powerful device can be of great benefit to both.

Note

[1] Card, Moran and Newell present variability data only in terms of the coefficient of variability, sd/mean (sd = standard deviation). Unfortunately, this index does not have additive properties, so one cannot use it directly to analyze or compare

component sources of variation. The estimate given here was made by a crude reconstruction of the underlying variances, assuming the times for error and error-free components to be uncorrelated.

References

Barnard, P., Hammond, N., Maclean, A., and Morton, J. (1982). Learning and remembering interactive commands in a text-editing task. *Behavior and Information Technology,* 1, 347–358.

Black, J. B., and Sebrechts, M. M. (1981). Facilitating human-computer communications. *Applied Psycholinguistics,* 2, 149–177.

Card, S. K., English, W. K., and Burr, B. J. (1978). Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics,* 21, 601–613.

Card, S. K., Moran, T. P., and Newell, A. (1983). *The Psychology of Human-Computer Interaction,* Hillsdale, NJ, Lawrence Erlbaum.

Carroll, J. M. (1982). Learning, using and designing file-names and command paradigms. *Behavior and Information Technology,* 1, 327–346.

Egan, D. E., and Gomez, L. M. (1985). Assaying, isolating, and accommodating individual differences in learning a complex skill. In R. Dillon (Ed.), *Individual Differences in Cognition,* Vol. 2, New York, Academic Press.

Ehrenreich, S. L. (1985). Computer abbreviations: evidence and synthesis. *Human Factors,* 27, 143–155.

Ehrenreich, S. L., and Porcu, T. A. (1982). Abbreviations for automated systems: Teaching operators the rules. In A. Badre and B. Schneiderman (Eds.) *Directions in Human Computer Interaction,* Norwood, NJ, Ablex, pp. 111–135.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling amplitude of movement. *Journal of Experimental Psychology,* 47, 381–391.

Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. (1983). Statistical semantics: Analysis of the potential performance of keyword information systems. *Bell System Technical Journal,* 62, 1753–1806. Also in J. C. Thomas and M. Schneider (Eds.), *Human Factors and Computer Systems,* Norwood, NJ, Ablex, 1983.

Gomez, L. M., and Lochbaum, C. C. (1984). People can retrieve more objects with enriched key-word vocabularies. But is there a human performance cost? *Proceedings of Interact 1984,* Amsterdam, Elsevier.

Goss, A. E., and Nadine, C. F. (1965). *Paired-Associate Learning. The Role of Meaningfulness, Similarity and Familiarization,* New York, Academic Press.

Gould, J. D., Conti, J., and Hovanyecz, T. (1983). Composing letters with a simulated listening typewriter. *Communications of the ACM, 26*, 295–308.

Grudin, J., and Barnard, P. (1984). The cognitive demands of learning and representing names for text editing. *Human Factors, 26.*

Hirsch-Pasek, K., Nudelman, S., and Schneider, M. L. (1982). An experimental evaluation of abbreviation schemes in limited lexicons. *Behavior and Information Technology*, 1, 359–369.

Landauer, T. K., and Galotti, K. A. (1984). What makes a difference when? Comments on Grudin and Barnard. *Human Factors, 26*, 423–429.

Landauer, T. K., and Nachbar, D. W. (1985). Selection from alphabetic and numeric menu trees using a touch screen: Breadth, depth and width. *CHI '85 Proceedings* (special issue of *SIGCHI*), New York, ACM.

Landauer, T. K., Galotti, K. A., and Hartwell, S. (1983). Natural command names and initial learning: A study of text-editing terms. *Communications of ACM, 26*, 495–503.

Ledgard, H., Whiteside, J. A., Singer, A., and Seymour, W. (1980). The natural language of interactive systems. *Communication of ACM*, 23, 110, 556–563.

Miller, D. P. (1981). The depth/breadth tradeoff in heirarchical computer menus. *Proceedings of the Human Factors Society*, pp. 296–300.

Miller, G. A. Galanter, E., and Pribram, K. H. (1960). *Plans and the Structure of Behavior*, New York, Holt, Rinehart and Winston.

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

Norman, D. A. (1981). The trouble with UNIX. *Datamation*, 27, 139–150.

Olson, J. R., and Nilsen, E. (1985). Analysis of the cognition involved in software interaction. Paper at 26th meeting, Psychonomic Society, Boston.

Payne, S. J., and Green, T. R. G. (1983). The user's perception of the interaction language: A two level model. In A. Janda (Ed.), *CHI '83 Conference Proceedings.*

Polson, P. G., and Kieras, D. E. (1985). A quantitative model of the learning and performance of text editing knowledge. In *CHI '85 Conference Proceedings* (special issue of *SIGCHI*), New York, ACM.

Postman, L. (1971). Transfer, interference and forgetting. In J. W. Kling and L. A. Riggs (Eds.), *Woodworth and Schlosberg's Experimental Psychology*, 3rd edition, New York, Hoft, Rinehart and Winston.

Reisner, P. (1981). Formal grammer and human factors design of an interactive software system. *IEEE Transactions on Software Engineering*, SE-7, 229–240.

Scapin, D. L. (1981). Computer commands in restricted natural language: Some aspects of memory of experience. *Human Factors,* 23, 365–375.

Snowberry, K., Parkinson, S. R., and Sisson, N. (1983). Computer display menus. *Ergonomics,* 26, 699–712.

Streeter, L. A., Ackroff, J. M., and Taylor, G. A. (1983). On abbreviating command names. *The Bell System Technical Journal,* 62, 1807–1828.

Tullis, T. S. (1985). Designing a menu-based interface to an operating system. *CHI '85 Proceedings* (special issue of *SIGCHI*), New York, ACM.

Underwood, B. J., and Schultz, R. W. (1960). *Meaningfulness and Verbal Learning,* Philadelphia, Lippincott.

Welford, A. T. (1980). Choice reaction time: Basic concepts. In A. T. Welford (Ed.), *Reaction Time,* New York, Academic.