1. An Applied Information-Processing Psychology

A scientific psychology should not only help us to understand our own human nature, it should help us in our practical affairs. In educating our children, it should help us to design environments for learning. In building airplanes, it should help us to design for safety and efficiency. In staffing for complex jobs, it should help us to discover both the special skills required and those who might have them. And on and on. Given the breadth of environments we design for ourselves, there is no limit to the number of domains where we might expect a scientific knowledge of human nature to be of use.

The domain of concern to us, and the subject of this book, is how humans interact with computers. A scientific psychology should help us in arranging this interface so it is easy, efficient, error-free—even enjoyable.

Recent advances in cognitive psychology and related sciences lead us to the conclusion that knowledge of human cognitive behavior is sufficiently advanced to enable its applications in computer science and other practical domains. The years since World War II have been the occasion for an immense wave of new understandings and new techniques in which man has come to be viewed as an active processor of information. In the last decade or so, these understandings and techniques have engulfed the main areas of human experimental psychology[1]: perception,[2] performance,[3] memory,[4] learning,[5] problem solving,[6] psycholinguistics.[7] By now, cognitive psychology has come to be dominated by the information-processing viewpoint.

A major advance in understanding and technique brings with it, after some delay, an associated wave of applications for the new knowledge. Such a wave is about to break in psychology. The information-processing view will lead to a surge of new ways for making psychology relevant to our human needs. Already the concepts of information-processing psychology have been applied to legal eyewitness testimony[8] and to the design of intelligence tests.[9] And in the study of man-machine systems and engineering psychology, it has for some time been common to include a block diagram of the overall human information-processing system in the introductory chapter of textbooks,[10] even though the reach of that block diagram into the text proper is still tenuous. There are already the beginnings of a subfield, for which various names (associating the topic in different ways) have been suggested: user sciences,[11] artificial psycholinguistics,[12] cognitive ergonomics,[13] software psychology,[14] user psychology,[15] and cognitive engineering.[16]

[1] For representative examples see Lindsay and Norman's (1977) *Human Information Processing*, Anderson's (1980) *Cognitive Psychology and its Implications*, the *Handbook of Learning and Cognitive Processes* (Estes, ed. 1975-1978), the *Attention and Performance* collections of papers (Kornblum, 1973; Rabbitt and Dornič, 1975; Dornič, 1977; Requin, 1978; Long and Baddeley, 1981), and the journal *Cognitive Psychology*.

[2] Examples: Broadbent (1958), *Perception and Communication*; Green and Swets (1966), *Signal Detection Theory and Psychophysics*; Neisser (1967), *Cognitive Psychology*; Cornsweet (1970) *Visual Perception*.

[3] Examples: Fitts and Posner (1967), *Human Performance*; Welford (1968), *Fundamentals of Skill*; Kintsch (1974), *The Representation of Meaning in Memory*; Tversky (1977), "Feature of similarity"; Posner (1978), *Chronometric Explorations of the Mind*.

[4] Examples: Anderson and Bower (1973), *Human Associative Memory*; Baddeley (1976), *The Psychology of Memory*; Crowder (1976), *Principles of Learning and Memory*; Murdock (1974), *Human Memory, Theory and Data*.

[5] Examples: Fitts (1964), "Perceptual-motor skill learning"; Klahr and Wallace (1976), *Cognitive Development: An Information-Processing View*; Anderson (1981a), *Cognitive Skills and their Acquisition*.

[6] Example: Newell and Simon (1972), *Human Problem Solving*.

Our own goal is to help create this wave of application: to help create an applied information-processing psychology. As with all applied science, this can only be done by working within some specific domain of application. For us, this domain is the human-computer interface. The application is no offhand choice for us, nor is the application dictated solely by its extrinsic importance. There is nothing that drives fundamental theory better than a good applied problem, and the cognitive engineering of the human-computer interface has all the markings of such a problem, both substantively and methodologically. Society is in the midst of transforming itself to use the power of computers throughout its entire fabric—wherever information is used—and that transformation depends critically on the quality of human-computer interaction. Moreover, the problem appears to have the right mixture of industrial application and symbol manipulation to make it a "real-world" problem and yet be within reasonable reach of an extended cognitive psychology. In addition, we have personal disciplinary commitments to computer science as well as to psychology.

This book reports on a program of research directed towards understanding human-computer interaction, with special reference to text-editing systems. The program was undertaken as an initial step towards the applied information-processing psychology we seek. Before outlining individual studies, it is appropriate to sketch how this effort fits in with the larger endeavor.

[7] Example: Clark and Clark (1976), *Psychology and Language: An Introduction to Psycholinguistics*.

[8] Loftus (1979).

[9] Hunt, Frost, and Lunneborg (1973).

[10] Sheridan and Ferrell (1974); McCormick (1976).

[11] Vallee (1976).

[12] Sime and Green (1974).

[13] Sime, Fitter, and Green (1975).

[14] Shneiderman (1980).

[15] Moran (1981*a*).

[16] Norman (1980).

1.1. THE HUMAN-COMPUTER INTERFACE

The human-computer interface is easy to find in a gross way—just follow a data path outward from the computer's central processor until you stumble across a human being (Figure 1.1). Identifying its boundaries is a little more subtle. The key notion, perhaps, is that the user and the computer engage in a communicative dialogue whose purpose is the accomplishment of some task. It can be termed a dialogue because both the computer and the user have access to the stream of symbols flowing back and forth to accomplish the communication; each can interrupt, query, and correct the communication at various points in the process. All the mechanisms used in this dialogue constitute the interface: the physical devices, such as keyboards and displays, as well as computer's programs for controlling the interaction.

At any point in the history of computer technology there seems to be a prototypical user interface. A few years ago it was the teletypewriter; currently it is the alphanumeric video-terminal. But the actual diversity is now much greater. All so-called "remote entry" devices count as interfaces; and a large number of such specialized devices exist in the commercial and industrial world to record sales, maintain inventory records, or control industrial processes. Almost all such devices are fashioned from the same basic sorts of components (keyboards, buttons, video displays, printers) and connect to the same sorts of information-processing mechanisms (disks, channels, interrupt service routines).

The very existence of the direct human-computer interface is itself an emergent event in the development of computers. If we go back twenty years, the dominant scheme for entering information into a computer consisted of a trio of people. First there was the user, someone who wanted to accomplish some task with the aid of the computer. The user encoded what he wanted onto a coding sheet, then sent it to a second person, the keypunch operator, who used an off-line device, the keypunch, to create a deck of punched cards that encoded the same information in a different form. The cards in turn went to a third person, the computer-operator, who entered the cards into the computer via the card reader. The computer then responded by printing messages and data on paper for the operator to gather up and send back to the user. The relationship between the user and the computer was sufficiently remote that it should be likened more to a literary correspondence than to a conversational dialogue. It is the general

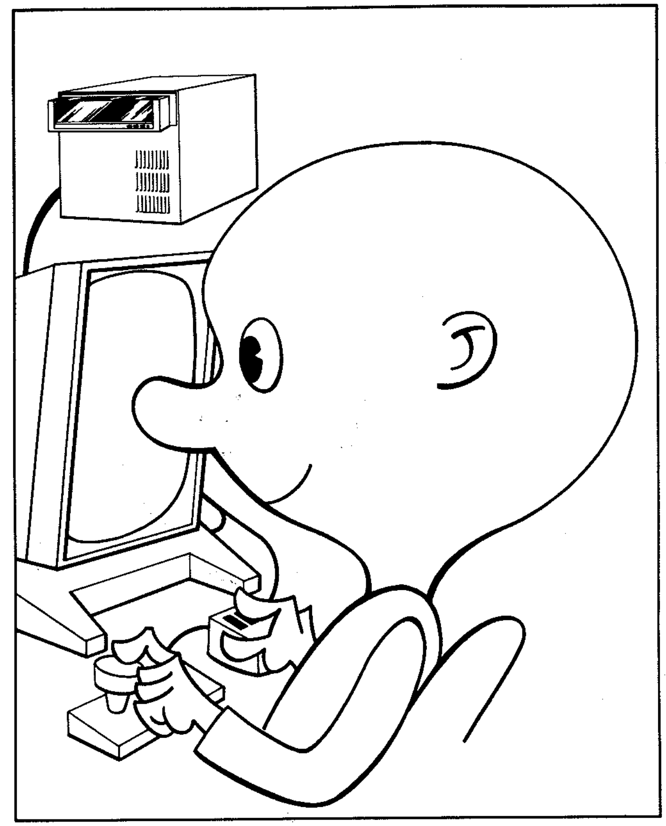

Figure 1.1. The human-computer interface.

demise of such arrangements involving human intermediaries, and the resultant coupling of the user directly to the computer, that has given rise to the contemporary human-computer interface. Whatever continued evolution the interface takes—and it will be substantial—human-computer interaction is unlikely ever to lose this character of a conversational dialogue.

Of course, there is much more to improving computer interfaces than simply making them conversational. Informal evidence from the direct experience of users provides numerous examples of current interface deficiencies:

> In one text-editing system, typing the word *edit* while in command mode would cause the system to select every-

> thing, *d*elete everything, and then *i*nsert the letter *t* (this last making it impossible to use the system Undo command to recover the deleted text because only the last command could be undone).
>
> In another text-editing system, so many short commands were defined that almost any typing error would cause some disaster to happen. For example, accidentally typing CONTROL-E would cause the printer to be captured by the user. Since no indication of this event was given, no other users would be permitted to print until the other users eventually discovered who had the printer. In an even more spectacular instance, accidentally typing CONTROL-Z would delete all the user's files—permanently.
>
> In one interactive programming system, misspelling a variable name containing hyphens (a common way of marking off parts of a name) would cause the system to rewrite the user's program, inserting code to subtract the parts of the name. In many cases, the user would have to mend his program by hand, laboriously searching for and editing the damaged code.
>
> In a set of different subsystems meant to be used together, the name "List" was given to many different commands, each having a different meaning: (1) send a file to the printer to make a hardcopy, (2) show the directory of files on the display, (3) show the content of a file on the display, (4) copy the workspace to a file, (5) create a particular kind of data structure.

Yet, when one looks at the teletype interfaces of yesterday, it is clear that substantial progress has been made. The emergence of the direct human interface, circumventing the keypuncher and operator, must itself be counted as an improvement of enormous value. We now have interfaces that allow the use of computers for such highly interactive tasks as making engineering drawings and taking airline reservations. But despite considerable advancements, the systems we have are often ragged and in places are sufficiently poor to cripple whole ranges of use.

What strikes one most noticeably about existing interfaces, besides all the little ways they fail, is that their failures appear to be unnecessary. Why, when interaction could be so smooth, even elegant, is it often so rough, even hazardous? Two observations may help explain this perplexing state of affairs.

First, interaction with computers is just emerging as a human activity. Prior styles of interaction between people and machines—such as driver and automobile, secretary and typewriter, or operator and control room—are all extremely lean: there is a limited range of tasks to be accomplished and a narrow range of means (wheels, levers, and knobs) for accomplishing them. The notion of the *operator* of a machine arose out of this context. But the user is not an operator. He does not operate the computer, he communicates with it to accomplish a task. Thus, we are creating a new arena of human action: communication *with* machines rather than operation *of* machines. What the nature of this arena is like we hardly yet know. We must expect the first systems that explore the arena to be fragmentary and uneven.

Second, the radical increase in both the computer's power and its performance/cost ratio has meant that an increasing amount of computational resources have become available to be spent on the human-computer interface itself, rather than on purely computational tasks. This increase of deployable resources exacerbates the novelty of the area, since entirely new styles of interaction become available coincidentally with an increased amount of computational ability available per interaction. These new styles often lead to completely new interfaces, which are then even more ragged than before. At the same time, opportunities for the invention of good interfaces also increase rapidly, accounting for the leaps and bounds we have seen in terms of major improvements in functionality and ease of use.

1.2. THE ROLE OF PSYCHOLOGY

Many in the computer field agree that there is an obvious way to design better human-computer interfaces. Unfortunately, they disagree on what it is. It is obvious to some that psychological knowledge should be applied. Their slogan might be, in the words of Hansen (1971): "Know the user!" It is obvious to others that the interface should simply be designed with more care—that if designers were given the goal of good interfaces, rather than stringent cost limits or tight deadlines, then they would produce good designs. Their slogan might be: "Designers are users too—just give them the time and freedom to design it right!" And it is obvious to others still that one should pour the effort into some new components—flat displays, color graphics, or dynamically codeable microprocessors in the terminal. Their slogan might be: "Make the components good enough and the system will take care of itself!"

Who is to gainsay each of these their point? The technology limits, often severely, what can be done. All the human engineering in the world will not turn a 10-character-per-second teletypewriter into a high-resolution graphics terminal. The history of terminal development so far is writ largely in terms of advances in basic interface components, most notably the resources to allow substantial computational cycles to be devoted to the interface. It is easy to point to current limitations whose lifting will improve the interface by orders of magnitude. Immense gains will occur when the display holds not the common 24 × 80 characters (the typical alphanumeric video terminal, widely available today), but a full page of 64 × 120 characters (the typical 1000 × 800 pixel video terminal, available at a few places today), or even the full drafting board of 512 × 512 characters (not really available anywhere, yet, as far as we know).

Moreover, any accounting will have to credit the majority of the capabilities and advances at the interface to design engineers and only a few of them to psychologists. However many imperfections there remain in the interface, the basic capabilities and inspired creations that do exist came out of an engineering analysis of the functions needed and the fact that the designer, being human, could empathize directly with the user.

And yet, there remain the mini-horror stories—of systems where, after the fact, it became clear that either the nature or the limitations of the user were not appreciated, and some design foolishness was committed. Since it is these stories that come to mind in discussing the role of the human at the interface, it is often assumed that all that one needs are ways of checking to be sure that the obvious is not overlooked; "All we need from psychology is a few good checklists!" might be the slogan here. But as we shall see, there is more to human-computer interaction than can be caught with checklists.

The role psychology might be expected to play in the design of the user-computer interface is suggested by the results it was able to achieve for military equipment during World War II. At that time, it had become apparent that a strong limiting factor in realizing the potential of man-machine systems, such as radar sets and military aircraft, lay in the difficulty of operating the equipment. Out of a wartime collaboration between natural scientists, engineers, and psychologists came major advances, not only with respect to the man-machine systems being designed, but also with respect to psychological theory itself. Examples of the latter include the theory of signal detection, manual control theory, and a methodology for the design of cockpit instrument displays. That with psychological attention to human performance airplanes became more flyable encourages us to believe that with psychological attention to human performance computers can become more usable.

1.3. THE FORM OF AN APPLIED PSYCHOLOGY

What might an applied information-processing psychology of human-computer interfaces be like and how might it be used? Imagine the following scenario:

> A system designer, the head of a small team writing the specifications for a desktop calendar-scheduling system, is choosing between having users type a key for each command and having them point to a menu with a lightpen. On his whiteboard, he lists some representative tasks users of his system must perform. In two columns, he writes the steps needed by the "key-command" and "menu" options. From a handbook, he culls the times for each step, adding the step times to get total task times. The key-command system takes less time, but only slightly. But, applying the analysis from another section of the handbook, he calculates that the menu system will be faster to learn; in fact, it will be learnable in half the time. He has estimated previously that an effective menu system will require a more expensive processor: 20% more memory, 100% more microcode memory, and a more expensive display. Is the extra expenditure worthwhile? A few more minutes of calculation and he realizes the startling fact that, for the manufacturing quantities anticipated, training costs

for the key-command system will exceed unit manufacturing costs! The increase in hardware costs would be much more than balanced by the decrease in training costs, even before considering the increase in market that can be expected for a more easily learned system. Are there advantages to the key-command system in other areas, which need to be balanced? He proceeds with other analyses, considering the load on the user's memory, the potential for user errors, and the likelihood of fatigue. In the next room, the Pascal compiler hums idly, unused, awaiting his decision.

The system designer is engaged in a sort of psychological civil engineering, trading computed parameters of human performance against cost and other engineering variables. The psychological science base necessary to make possible his design efforts is the sort of applied psychology that is the topic of this book. Such a psychology must necessarily be homogeneous in form with the rest of the engineering science base to allow tradeoffs between psychological and other design considerations. To be useful, we would argue, such a psychology must be based on task analysis, calculation, and approximation.

Task Analysis. When psychology is applied in the context of a specific task, much of the activity hardly seems like psychology at all, but rather like an analysis of the task itself. The reason for this is clear: humans behave in a goal-oriented way. Within their limited perceptual and information-processing abilities, they attempt to adapt to the task environment to attain their goals. Once the goals are known or can be assumed, the structure of the task environment provides a large amount of the predictive content of psychology.

Calculation. The ability to do calculations is the heart of useful, engineering-oriented applied science. Without it, one is crippled. Applications are, of course, still possible, as witness mental testing, behavior modification, assertiveness training, and human-factors investi-gations of display readability. But what is needed to support an engineering analysis are laws of parametric variation, applicable on the basis of a task analysis.

Psychology is not strong on calculation, though a few useful laws, such as Power Law of Practice, exist. The reason might be thought to be an inherent characteristic of psychology, or maybe even more generally, of all human sciences. Our view is the opposite. Psychology

is largely non-calculational because it has followed a different drummer. It has been excessively concerned with hypothesis testing—with building techniques to discriminate which of two ideas is right. If one changes what one wants from the science, one will find the requisite techniques. Interestingly, a branch of the human sciences, work-measurement industrial engineering, indeed asked a different question—namely, how long would it take people to do preset physical tasks—and it obtained useful answers.

Approximation. If calculations are going to be made rapidly, they are necessarily going to be over-simplified. Nature—especially human nature—is too complex to be written out on the back of an envelope. But in engineering, approximations are of the essence. It is vital to get an answer good enough to dictate the design choice; additional accuracy is gilding the computational lily.

Again, psychology has in general not asked after approximations, though it has certainly learned to talk in terms of simplified models. The neglect of approximation has been especially encouraged by the emphasis on statistical significance rather than on the magnitude of an effect. A difference of a few percent in performance at two levels of an independent variable is usually of little practical importance and can often be ignored in an approximation, even if the difference is highly significant statistically. But if there is no external criterion—no design decision to be made, for instance—then there is no way to tell which approximations are sufficient.

But, whereas an applied psychology of human-computer interaction should be characterized by task analysis, calculation, and approximation, these are not the only considerations. It is obvious that an applied psychology intended to support cognitive engineering should also be relevant to design. It is less obvious, but nonetheless true, that to be successful, an applied psychology should be theory-based.

RELEVANT TO DESIGN

Design is where the action is in the human-computer interface. It is during design that there are enough degrees of freedom to make a difference. An applied psychology brought to bear at some other point is destined to be half crippled in its impact.

We suspect that many psychologists would tend to pick evaluation as the main focus for application (though some might have picked training). Evaluation is what human factors has done best. Given a real system, one can produce a judgment by experimentation. Thus, the main tool in the human-factors kit has been the methodology of experimental design, supported by concomitant skill in experimental control and in statistics with which to assess the results. The emphasis on evaluation is widespread: There is a whole subfield of psychology whose concern is to evaluate social action programs. The testing movement is fundamentally evaluational in character, whether concerned with intelligence testing or with clinical assessment.

Applying psychology to the evaluation of systems is assuredly easier than applying it to the design of systems. In evaluation, the system is given; all its parts and properties are specified. In design, the system is still largely hypothetical; it is a class of systems. On the other hand, there is much less leverage in system evaluation than in system design. In design, one wants results expressed explicitly as a function of some controllable parameters, in order to explore optimization and sensitivity. In evaluation, this urge is much diminished; experimental evaluation is so expensive as to be prohibitive, permitting exploration of only two or three levels of each independent variable. Most importantly, by the time a system is running well enough to evaluate, it is almost inevitably too late to change it much. Thus, an applied psychology aimed exclusively at evaluation is doomed to have little impact.

There are several choices for how to institutionalize an applied psychology. First, psychologists could be the primary professionals in the field. Though possible in some fields, such as mental health, counseling, or education, we think this arrangement unlikely for computers. The field is already solely in the possession of computer engineers and scientists. Second, psychologists could be specialists, either as members of separate human-factors units within the organizations or as another individual specialty within the primary design team. Our reasons for not favoring separate psychology units reflect the additional separation we believe they imply between the psychology and the development of interfaces. Application of psychology would shift too strongly towards evaluation and away from the main design processes.

We favor a third choice: that the primary professionals—the computer system designers—be the main agents to apply psychology. Much as a civil engineer learns to apply for himself the relevant physics of bridges, the system designer should become the possessor of the relevant applied psychology of human-computer interfaces. Then and only then will it become possible for him to trade human behavioral considerations against the many other technical design considerations of system configuration and implementation. For this to be possible, it is necessary that a psychology of interface design be cast in terms homogeneous with those commonly used in other parts of computer science and that it be packaged in handbooks that make its application easy. Thus, the system designer in our scenario finds the design handbook more efficient to use than plunging blindly into code with his Pascal compiler, although he may still find it profitable to engage in exploratory implementation.

THEORY-BASED

An applied psychology that is theory-based, in the sense of articulating a mechanism underlying the observed phenomena, has advantages of insight and integration over a purely empirical approach. The point can be made by reference to two examples of behavioral science lacking a strong theory in this sense: work-study industrial engineering, referred to earlier, and intelligence testing. Rather than develop the theory of skilled movement, the developers of the several movement time systems chose an empirical approach, tabulating the times to make various classes of movements and ignoring promising theoretical developments such as Fitts's Law (at least until recently). Although their tables of motion times ran to four significant figures, they ignored the variance of the times and interactions between sequential motions, thus rendering the apparent precision illusory. This lack of adequate theoretical development made the work, despite its impressive successes, vulnerable to attacks from outside the field (see Abruzzi, 1956; Schmidtke and Stier, 1961). Similarly, in mental testing, the lack of a psychological theory of the mental mechanisms underlying intelligence (as opposed to a purely statistical theory of test construction) has put the validity of mental tests in doubt despite, again, impressive successes.

It is natural for an applied psychology of human-computer interaction to be based theoretically on information-processing psychology, with the latter's emphasis on mental mechanism. The use of models in which man is viewed as a processor of information also provides a common framework in which models of memory, problem solving, perception, and behavior all can be integrated with one another. Since the system designer also does his work in information-processing terms, the emphasis is doubly appropriate. The lack of this common framework is one reason why it would be difficult to meld in important techniques such as the use of Skinnerian contingent reinforcement. It is not that the techniques are not useful in general, nor that they cannot be applied to the problems of the human-computer interface; but within the framework that underlies this book, they would show up as isolated techniques.

The psychology of the human-computer interface is generally individual psychology: the study of a human behaving within a non-human environment (though, interestingly, interacting with another active agent). But within the study all psychological functioning is included—motor, perceptual, and cognitive. Whereas much psychology tends to focus on small micro-tasks studied in isolation, an applied psychology must dwell on the way in which all the components of the human processor are integrated over time to do useful tasks. For example, it might take into account interactions among the following: the ease with which commands can be remembered, the type font of characters as it affects legibility of the commands, the number of commands in a list, and anything else relevant to the particular interface. The general desirability of such wide coverage has never been in doubt. It appears in our vision of an applied psychology because wide coverage, especially the incorporation of cognition, now seems much more credible than it did twenty years ago. On the other hand, motivational and personality issues are not included. Again, there is hardly any doubt of the desirability of including them in an applied psychology, but it is unclear how to integrate the relevant existing knowledge of these topics.

1.4. THE YIELD FOR COGNITIVE PSYCHOLOGY

The textbook view is that as a science develops it sprouts applications, that knowledge flows from the pure to the applied, that the backflow is the satisfaction (and support) that comes to a science from benefiting society. We have been reminded often enough that such a view does violence to the realities in several ways. Applied domains have a life and source of their own, so that many ingenious applications do not spring from basic science, but from direct understanding of the task in an applied context—from craft and experience. More importantly in the present context, applied investigations vitalize the basic science; they reveal new phenomena and set forth clearly what it is that needs explanation. The mechanical equivalent for heat, for instance, arose from Count Rumford's applied investigations into the boring of brass cannon; and the bacteriological origin of common infectious diseases eventually arose, in part, out of studies by Pasteur on problems besetting the

fermentation of wine. The basic argument was made for psychology by Bryan and Harter (1898); and numerous applied psychological models exist to remind us of what is possible (for example, Bryan and Harter's 1898 and 1899 studies of telegraphy, Book's 1908 studies of typewriting, and Dansereau's 1968 study of mental arithmetic).

These general points certainly hold for an applied cognitive psychology, and on the same general ground that they hold for all sciences. However, it is worth detailing the three main yields for cognitive psychology that can flow from a robust applied cognitive psychology.

The first contribution is to the substance of basic cognitive psychology. The information-processing revolution in cognitive psychology is just beginning. Many domains of cognitive activity have hardly been explored. Such explorations are not peripheral to the basic science. It is a major challenge to the information-processing view to be able to explain how knowledge and skill are organized to cope with all kinds of complex human activities. Each application area in fact becomes an arena in which new problems for the basic science can arise. Each application area successfully mastered offers lessons about the ways in which the basic science can be extended to cover new areas. Ultimately, as a theory becomes solidified, application areas contribute less and less to the basic science. But at the beginning, just the reverse is true.

The domain of human-computer interaction is an example of such an unexplored domain. It has strong skill components. People who interact with computers extensively build up a repertoire of efficient, smooth, learned behaviors for carrying out their routine communicative activities. Yet, the interaction is also intensely cognitive. The skills are wielded within a problem-solving context, and the skills themselves involve the processing of symbolic information. As we shall see in abundance, even the most routine of these activities, such as using a computer text-editing program, requires the interpretation of instructions, the formulation of sequences of commands, and the communication of these commands to the computer.

The second contribution is to the style of cognitive psychology rather than to its substance. We believe that the form of the psychology of human-computer interaction, with its emphasis on task analysis, calculation, and approximation, is also appropriate for basic cognitive psychology. The existing emphasis in psychology on discriminating between theories is certainly understandable as a historical development.

However, it stifles the growth of adequate theory and of the cumulation of knowledge by focusing the attention of the field on the consequences of theories, however uninteresting in themselves, that can be used to tell whether idea A or idea B is correct. Measurements come to have little value in themselves as a continually growing body of useful quantitative knowledge of the phenomena. They are seen instead primarily as indicators fashioned to fit the demands of each experimental test. Since there is no numerical correspondence across paradigms in what is measured, the emphasis on discrimination fosters a tendency towards isolation of phenomena in specific experimental paradigms.

The third contribution is simply that of being a successful application, though it sounds a bit odd to say it that way. Modern cognitive psychology has been developing now for 25 years. If information-processing psychology represents a successful advance of some magnitude, then ultimately it must both affect the areas in which psychology is now applied and generate new areas of application.

1.5. THE YIELD FOR COMPUTER SCIENCE

It is our strong belief that the psychological phenomena surrounding computer systems should be part of computer science. Thus, we see this book not just as a book in applied psychology, but as a book in computer science as well. When university curriculum committees draw up a list of "what every computer scientist should know to call himself a computer scientist," we think models of the human user have a place alongside models of compilers and language interpreters.

The fundamental argument is worth stating: Certain central aspects of computers are as much a function of the nature of human beings as of the nature of the computers themselves. The relevance of both computer science and psychology to the design of programming languages and the interface is easy to argue, but psychological considerations enter into more topics in computer science than is usually realized. The presumption that has governed two generations of operating systems, for instance, that time-sharing systems should degrade response time as the number of users increases, is neither dictated by technology nor independent of the psychology of the user. A sufficiently crisp model of the effects of such a feature on the user could have turned the course of development of operating systems into quite different channels of development (into the logic of guaranteed service, contracted service, or proportionately graded services, for example). The yield for computer science that can flow from an applied psychology of human-computer interaction is engineering methods for taking the properties of users into account during system design.

1.6. PREVIEW

In this book, we report on a series of studies undertaken to understand the performance of users on interactive computing systems. Since new knowledge and insight are often achieved by first focusing on concrete cases and then generalizing, we direct a major portion of our effort towards user performance on computer text-editing systems. From this beginning, we try to generalize to other systems and to cognitive skill generally. We address four basic questions: (1) How can the science base be built up for supporting the design of human-computer interfaces? (2) What are user performance characteristics in a specific human-computer interaction task domain, text-editing? (3) How can our results be cast as practical models to aid in design? (4) What generalizations arise from the specific studies, models, and applications?

SCIENCE BASE

Chapter 2 begins by discussing the existing scientific base on which to erect an applied psychology of the human-computer interface. It does not review all the sources in their own terms—what is available from cognitive psychology, human factors, industrial engineering, manual control, or the classical study of motor skills—rather, it lays out a model of the human information-processor that is suited to an applied psychology and justified by current research.

TEXT-EDITING

Attention then turns to a detailed examination of text-editing as a prototypical example of human-computer interaction. An elementary requirement for understanding behavior at the interface is some gross quantitative information about user behavior, to provide a background picture against which to place more detailed studies in context. The three studies in Chapters 3 and 4 provide such a picture. Two of these (Chapter 3), a benchmark study comparing text-editing systems and a study of the individual user differences, allow one to assess the variability in performance time arising from editing system design and from individual user differences. The third study (Chapter 4) uses the data of Chapter 3 to explore how well a simple model, in which all editing modifications are assumed to take the same time, does at analyzing tradeoffs between using a computer text-editor vs. using a typewriter.

The next three chapters develop an information-processing model for the behavior of users with an editing system. Chapter 5 introduces the basic theory. The user is taken to employ goals, operators, methods, and selection rules for the methods (the GOMS analysis) to accomplish an editing task from a marked-up manuscript. Experimental verification of the analysis is given, and the effect on accuracy due to the detail with which the analysis is applied is also investigated. The routine use of an editing system is discussed as an instance of cognitive skill. Chapter 6 extends the model in three ways. First, the model is reduced to a complete, running computer simulation of user performance. Second, the analysis is extended to user behavior on a display-oriented system. Third, stochastic elements are introduced into the model to predict the distributions of performance times. Chapter 7 examines in detail one suboperation of editing: selecting a piece of text. Four different devices for doing this are tested, and a theoretical account is given for their performance.

ENGINEERING MODELS

Chapters 8 and 9 focus on the ways in which the GOMS analysis can be simplified to provide practical models for predicting the amount of time required by a user to do a task. In Chapter 8, a model at the level of individual keystrokes is presented that is sufficiently simple and accurate to be a design tool. The model is validated over several systems, tasks, and users; and examples are given for ways in which the model could be used in engineering applications. In Chapter 9, a second simplification of the GOMS analysis, this time at a more gross level, is presented This model is suited for cases where, as in the early stages of design, the system to be analyzed is not fully specified.

EXTENSIONS AND GENERALIZATIONS

So far, the studies have focused mostly on manuscript editing and on similar tasks where the user carries out a set of instructions. Chapter 10 extends the same kind of analysis to a particular problem-solving activity: the use of a computer system to lay out a VLSI electronic circuit. The analysis shows that the user behavior exhibits many of the characteristics of manuscript editing and that the behavior is indeed a routine cognitive skill, partially understandable in terms of the concepts already introduced.

Chapter 11 attempts to place results from the above studies in a larger theoretical context. It continues the discussion of text-editing as an instance of cognitive skill and the relationship between cognitive skill generally and problem solving. Chapter 12 addresses the role of psychological studies in design. It is argued that psychological studies should emphasize the creation of performance models. The several methods of doing this are discussed and provide a framework for summarizing the thrust of the present book. A number of guidelines for systems development that arise from our studies are listed.