11 Theory and Practice from Cognitive Science.

Michael Wilson
Central Laboratory to the Research Councils
Rutherford Appleton Laboratory

Abstract

This Chapter discusses the cognitive science approach to HCI, and in particular how some cognitive science models can be used to facilitate the goal of *User Interfaces for All*. Two major approaches to modeling human computer interaction are described, showing the benefits they offer, but also the limitations in their use by system developers. Two tools which overcome these usability problems, at the cost of constraining the cognitive phenomena they can capture, are also described. Finally, recent developments in computing technology and cognitive science are outlined, which could work synergistically to create future computer interfaces that are suitable for the broadest possible end user population.

1. Introduction

Cognitive science is concerned with the understanding of mental life, and the expression of that understanding in the form of theories and models. It develops frameworks and models of human behavior, which can be used to provide insights into human computer interaction. Usually, these insights are either adopted into requirements capture or evaluation methods, or lead to user interface design ideas which require engineering into future computing interfaces (such as those discussed in chapter 6). However, cognitive science approaches are difficult to directly embed into tools which are usable by general designers, because background knowledge is required to select the appropriate models and to interpret results.

The models produced by cognitive science are designed to capture the generalities of a population and are poor at addressing individual differences, or small groups of specialized users, since they require statistically valid data to be collected in order to justify them. However, the general architectures provided by cognitive science have given rise to much of the work on individual differences (chapter 4), or user modeling (chapter 18) discussed elsewhere in this book.

This chapter will not review the whole cannon of work in cognitive science, nor even that subset related to human computer interaction (for a general review of the contribution of cognitive science and human information processing psychology to HCI see, e.g. Barnard, 1995). The next section discusses the cognitive science approach in order to lay a foundation for the later descriptions of how some of its models can be used within the *User Interfaces for All* enterprise. Section three describes two major approaches to modeling human computer interaction (TAL and ICS), showing the benefits they offer, but also the limitations in their usability. Section four outlines two tools which overcome these problems of usability for system developers, but at the cost of constraining the phenomena they can address (GOMS and KADS). Finally, recent developments in computing technology and cognitive science are outlined which could work synergistically to create future computer interfaces capable of being used by the broadest possible end user population, following the *User Interfaces for All* principles.

2 The Cognitive Science Approach

2.1 Methodology in Cognitive Science

The big questions about the structure of matter and the universe were moved beyond the arena of ecclesiastical and philosophical debate to be the subject of logical reasoning, supported by empirical results, in 1688, when Isaac Newton published his *Principia*. The big questions about the functioning of the human mind were not addressed by a similarly rigorous scientific method until 1879, when Wilhelm Wundt established the first laboratory of psychology at Leipzig.

Wundt reacted against the introspective method of the time so extremely, that he performed thousands of trials on users in each experiment, to ensure that his empirical results were replicable (Miller, 1966). Because of this rigor, he could only address local episodes of behavior which were amenable to such repeated measurement in detailed experiments, while more general phenomena remained the province of the introspectionists.

The tension between the desire to answer broad important questions about mental life, and the need to use arguments which draw upon methodologies that are sound, is a dialectic which drives the progress of psychology. Indeed this conflict has been present throughout the history of ideas, and has lead to swings between rationalism and romanticism. However, this chapter will only explore the recent developments in cognitive science, and the application of the results to human computer interaction.

The paradigm which dominated psychology through the first half of this century was *behaviorism*, which accepted that humans learn and act in the world as a result of stimuli which they encounter, and that a linking of those actions to the stimuli is all that can be validly argued. Behaviorism maintained a strong defense of scientific methods in psychology against introspectionism. However, by the 1960's, behaviorism was felt to have been too restrictive in the range of questions that it could address, and cognitive or information processing psychology had taken over as the dominant paradigm. Cognitive psychology permitted arguments to postulate mental processes and intermediate representations between stimuli and actions, upon which those processes could operate. . It also introduced methodologies to interpret the performance of humans in the execution of various tasks, with respect to the time taken and the number of errors made, in terms of these postulated processes and representations.

The methodology initially accepted as valid by cognitive psychology was restrictive, as illustrated by the interpretation, in terms of processes and representations, that could be proposed as a result of the time that people take to make decisions in reaction to stimuli (Sternberg, 1969). This logic proposes that variables whose effects are additive affect different processing stages, but two variables whose effects are interactive affect the same processing stage. For example, variables can be assigned a role in perceptual processing when they interact with marker variables whose effects can be assumed to be in perceptual processes, but not with variables whose effects are on response selection or action execution. Analyses based on the distributions of reaction times have been used to develop models of the stages of human information processing as boxes linked by arrows.

Although not as much as behavioral psychology, this methodology is still limited to interpreting overall task performance and does not permit observations of the intermediate stages of processes. Methodological developments in cognitive psychology were therefore directed at gaining information as to the intermediate states of processes, by looking at secondary actions performed during the performance of the primary task which would be effected by it. The most extreme methodology acceptable in cognitive psychology permits retrospective answers to questions about intermediate cognitive states during previous task performance as data suitable for validating theory. Various procedures for collecting verbal protocols while tasks are being performed have been suggested (e.g. Ericsson and Simon, 1984) which are accepted by the discipline with intermediate weight as validating evidence. However, their production may interfere with the performance of the primary task, and they may be rationalizations of behavior rather than accounts of intermediate processing.

This methodology is extreme for cognitive psychology in its closeness to introspection, but is clearly rigorous compared to the methodology of the deconstruction of cultural artifacts accepted in many humanities, where texts, paintings, films etc., can be deconstructed in terms of any of the major cultural distinctions (class conflict, feminism, racism, environmentalism, self/society or ego/superego, dominance/submission etc.). The extreme reaction against these overly free analytic methods in social science and the humanities has been to the equally extreme *ethnomethodology*, where talk, or artifacts, may merely be collected and presented. Any form of analysis or even juxtaposition is prohibited since it merely adds the interpretation of the analyst to those of the creator and reader/viewer, which amounts to creating the original artifact as well as the analysis underpinning it. To balance protocol analysis as rigorous deconstruction, a rigorous form of ethnomethodology, termed *situated action*, has been introduced into cognitive science (e.g. Norman, 1993). In the case of HCI, Suchman (1987) and Winograd and Flores (1986) argue that to develop better interfaces, we must focus on how people use them, rather than on how people think, or what computers can do. Although its origins are not within cognitive psychology, this approach has many similarities to the perception centered proposals of

Gibson (1979). Such a proposal argues that cognitive psychology should consider how humans perceive *affordances* in objects (e.g. a dial, a button) to act on them in particular ways (e.g. turn the dial, push the button), and that more direct links can be made between perception and action, than what can be facilitated by a single centralized representation. It is possible to view this as a cautious step back towards the constraints behaviorism imposed, in order to curb the freedom to theorize about intermediate representations.

When the postulation of mental processes and representations was first accepted, theory was required to be strictly tied to experimental data, and theories that were not strongly predictive of experimentally testable predictions were given little weight. If a prediction that a theory made was not experimentally supported, then the theory should be rejected, following the strict arguments of Popper (1992). Such structures were necessary in order to prevent the abuse of the freedom which was allowed by postulating mental processes and representations. These restrictions gradually mellowed, so that a layering arose in theoretical proposals between models as instances of theories, within general frameworks (e.g. Morton et al, 1979). Models make predictions which are strictly experimentally testable. Theories can give rise to sets of models, so if the predictions of one model failed, other predictions which arose from the same theory could still be held. Integrative frameworks provide a higher level structure which guides the integration of theories. Since integrative frameworks can combine theories accounting for data from different experimental paradigms, they are not in themselves testable by any single experimental paradigm. Rather, the predictions arising from set of integrated theories are tested for their ability to achieve a purpose, such as the engineering goals of HCI. However, there are always questions raised about the validity of integrating theories derived from different paradigms into a single framework. The cost of this increasing freedom to theorize, is an uncertainty as to methodological soundness, in both testability and the generalizability of results.

The trend towards further softening the methodological demands on theory led to the development of cognitive science from cognitive psychology in the mid 1970’s. Answers to broad questions concerning mental life require broad theories encompassing many aspects of that mental life, beyond performance on limited tasks. These are hard to motivate purely within the tasks addressed by experimental psychology. It was necessary to introduce theoretical findings from linguistics into theories, and to build theories whose complexity went far beyond the experimentally testable predictions arising from ‘box and arrow’ models. To build such theories, the modeling techniques used in artificial intelligence were also combined with methods from linguistics and cognitive psychology to form the discipline of cognitive science.

The merge of cognitive psychology and artificial intelligence introduces sound representational and computational modelling techniques to express cognitive psychology theories. Psychology has determined many empirical regularities and small scope theories to account for them, but it has been argued that to move forward it needs to search for a single unified theory of cognition as an alternative to integrative frameworks (Newell, 1990).

Although AI may provide the mechanism for this, it also has the potential to permit theorizing which is not sufficiently grounded in evidence for the purpose of psychology as a science.

The merge of cognitive psychology and linguistics can also cause problems in determining a sound methodology in cognitive science, arising from the diverging the methods the two disciplines use to generalize results to populations. The core methodology of cognitive psychology is to develop hypotheses which are tested by experiments on small samples of subjects. Results of experiments are subjected to statistical analyses. The power of this methodology in generalizing from the particular experiment to the population relies on the statistics to show that differences are significant for the population sampled. The sample can be taken from the population as a whole, or from sub-populations which are shown to differ on a secondary measure, correlated or interacting with the investigated phenomena. Thereby, individual differences among the population can be determined (see chapter 4 by Benyon). In contrast, the methodology commonly adopted linguistics is either to identify a phenomenon in the real world by a single example as a proof of existence, by relying on the judgment of a single individual as a native speaker of a language, or to abstract away from the performance of individuals to determine a generalization of a phenomenon to a universe of speakers - often represented as a grammar which captures the general competence of a population in a language, independently from their use in performance. These differing methodologies do not fit well together and lead to conflicts within cognitive science, particularly when considering variations between individuals in language related capacities and abilities, which are the noise to the linguist's general competence signal, yet can be the signal to the psychologist.

The distinctions drawn in this section between paradigm specific theories and integrative ones, between integrative theories and frameworks, between evaluation of theory for scientific validity or utility, and the relationships between experimentation on samples and variations in the population, are all drawn on in the description of the results of cognitive science modelling in section 3. The next section provides a brief mapping from cognitive science to the engineering discipline of human computer interaction.

2.2 The HCI problem in Cognitive Science terms

The objective of cognitive science is to understand mental life, and to express that understanding in the form of theories and models. In contrast, the objective of human-computer interaction comes from an engineering tradition, and is to design artifacts which facilitate and improve users' performance of tasks (e.g., computer use at work), or attainment of emotional states (e.g., computer use as entertainment). In order to perform design, an understanding of the users' performance of tasks is insightful, while specific tools which guide or evaluate designs are most immediately useful. To achieve these engineering objectives, it is necessary to: (i) identify those results of the Cognitive Science enterprise which are applicable in principle; (ii) transform these into a form which can be directly applied; and, (iii) hopefully reduce them to methods which can be adopted and executed by designers, without the need of considerable education in the disciplines which gave rise to them.

As described above, cognitive science promotes the use of explicit representations of knowledge and the reasoning processes of humans. The problem of HCI can be characterized in cognitive science terms as follows: given that a user wishes to perform a task in some domain of application, using a computer system as a tool, and communicate with the latter by establishing some form of dialogue, such a dialogue should be constructed in a way that maximizes effectiveness and efficiency of task performance, and thereby minimizes complexity of the communication between the user and the computer system. Users hold cognitive representations of the domain, task, computer system and dialogue which they reason over to take actions, and perform the task. In order to maximize the efficiency of task performance, we need to distribute processing between the user and system to match the abilities of each - sometimes called *distributed cognition*. In order to simplify the dialogue, we need to ensure that the system side is as compatible with the user's representations as possible, and as internally consistent as possible, so that users can identify the compatibility. In order to design computer systems with these properties, we need to model the domain, task, dialogue and user, and the system itself. We can use these models in three ways: firstly, in the development process to establish requirements, guide design decisions or establish evaluation methods to measure compatibility between the system's and user's representations; secondly, we can embody these models in documentation and training material to bring the user as close to the system as possible; or thirdly, we can embody these models in the system itself, allowing it to adapt to the user at run time. Of course, the initial scenario is too simple, since user's representations change over time through learning and forgetting. A task is not a disembodied goal, but is linked to higher motivational and emotional states. Users perform the task in an environmental context, where they may need to communicate with other humans who are involved in the performance of the task, through computer systems or by other means. Equally, single users perform many different tasks in different domains, while multiple users perform the same task. To address this last point in the spirit of *User Interfaces for All*, it is necessary for the models to capture the variation between users (and within individual users as they change over time). Models should also be applied in any of the three ways mentioned above, so that task performance can be as effective and efficient as possible for each individual user, rather than a stereotyped or average user.

"For the most part, useful theory is impossible, because the behavior of human-computer systems is chaotic or worse, highly complex, dependent on many unpredictable variables, or just too hard to understand." (Landauer, 1996). In the sense intended by Landauer, a single useful theory that can dictate system design characteristics to produce better designs than are possible through human skill and emulation is certainly not available, and probably impossible. However, as with other aspects of cognitive science, a large number of different models have been developed from different experimental paradigms, which address aspects of the solution. Although these models can not produce better designs, they can often provide insights into the options or trade-offs involved in design decisions, or provide constraints on those design options.

3. Applicable cognitive science theory

Three examples of applicable cognitive science theory are described in this section. The first example is paradigm specific, and theoretically weak, but very important in accounting for user variations in HCI which have consequences for design. The second moves to the theoretically stronger approach of a model (TAL), developed within an integrative theory, that captures changes during learning -. Finally, an integrative framework (ICS) is described, which can address the widest aspects of variation in HCI from a cognitive perspective, but is consequently underspecified in too many areas to be usable by system developers -. This trade-off between the detailed representation necessary to provide predictive power, and the usability of a modeling technique, is then further taken up in section four which describes two techniques, intended as tools for system developers.

3.1 Variation in users' domain language across the population

The exhortation, mentioned in the previous section, for design to be compatible with users' previous experience, is well founded (Barnard et al, 1981). To achieve such a compatibility in the terminology about the domain to be adopted, it appears straightforward to follow a linguistic competence approach in creating a dictionary of the terms typical of the specific domain, and use this dictionary in the system dialogue. However, considering the high degree of variation in vocabulary usage, compatibility can be impossible to achieve, and trying to produce it only leads to frustration for designers. An elegant series of studies (Furnas et al, 1982; Gomez and Lochbaum, 1984; Landauer, 1987) have been undertaken on a wide range of problem domains, including text editing, the contents of yellow pages directories, cooking, and goods "wanted/for sale" services. These studies show that people use a variety of descriptions when referring to the same items. Furthermore, they show that such a variability can be quantified, and, that the likelihood of any two people using the same term to refer spontaneously to the same concept ranges from 0.07 to 0.18. Therefore, any designer has a very low chance of choosing any single keyword which more than 20% of the population are likely to use if they can only use one. Statistical simulations based upon these data were used to explore the probable success of alternative access schemes, which showed that the probability of a successful match could be raised to 75-80% where the system accepts many terms for target information (Furnas et al, 1983).

These studies, involving considerable empirical evaluation and statistical modeling have brought about a reformulation of the design objective from selecting the name most compatible with the expectations of a population of users, to designing an efficient aliasing system, to capture the variation in the population's language use.

3.2 Variation in users' mental models during learning

Users construct mental models of the system as they learn about it. These models become richer during learning, and therefore users require different support as learning progresses. In order to develop interfaces suitable for the broadest possible end user population, at whatever stage of learning about a system users are, we need to understand, and model the learning process, applying the cognitive science method.

Halasz and Moran (1983) conducted an experiment in which half of a group were taught an explicit conceptual model of the calculator's stack, and the other half were not given a model but only taught how to perform the necessary tasks. No difference was found between the groups as for the accuracy in solving routine or even complex problems. However, when it came to solving problems which required some level of invention - that is, where the users had to invent new methods - the group taught the model got more problems right. In particular, this group was better able to perform novel tasks that required more cognitively intense problem solving. It was argued that the model helped users to construct a better problem space, in which to carry out the problem solving process necessary for creative solutions.

This example shows how different styles of teaching can produce different mental representations in users, which in turn result in different performance. A study by Wilson et al (1990), concerning users learning to use an office software suite, showed similar distinctions, due, not to the type of training, but to variations between users. In this study, users were trained initially on the core, or habitable subset, of functions in the office suite, which all users require to perform basic tasks. They were subsequently trained on more advanced functions. Performance tests on these and further untrained functions were given users in order to assess their skills and the transfer of learning, along with various off line measures of their knowledge of the system, including questions based on screen images, and questionnaires to assess verbalizable knowledge (either triggered *affordances* or conversational language). This study investigated a general view of users learning a mapping from a task to the actions required to perform it (the *task-action mapping* after Payne and Green, 1986) which is characterizable as a general model of skill acquisition (after Fitts, 1964). This model divides learning into three phases. In the first, users acquire sufficient fragments of knowledge about a system to support the performance of some tasks. In the second phase, the knowledge recruited to perform tasks is compiled into procedures. In the third phase, users draw on these compiled procedures, and exhibit a level of performance which is considered 'expert like'. A further distinction concerns the nature of the types of knowledge in the first and third phases. It is assumed (after Anderson, 1983) that the type of knowledge which accumulates in the first phase is accessible to the processes of verbalization, whereas the compiled procedures drawn on in the third phase would not be possible to articulate. During these changes in the representation of knowledge, there is a hypothesized parallel reduction in the time necessary to perform the task, and in the number of errors during performance. This change in performance time is called the 'power law of practice' which states that the time to perform a task decreases as a power law function of the number of times the task has been performed. It has recently been argued that this law does not only apply to the domain of motor skills, to which it was originally applied (Snoddy, 1926), but to the full range of human tasks (Newell and Rosenbloom, 1981), including perceptual tasks such as target detection (Neisser et al 1963), and purely cognitive tasks such as supplying justifications for geometric proofs (Neves and Anderson, 1981). Consequently, it is often assumed that as experience with a computer system increases, the general task performance time and error count reduces.

The progressive teaching of more sophisticated commands, used in the Wilson et al (1990) study, is similar to one advocated by Carroll and his colleagues under the banner of *training wheels* (e.g. Catrambone and Carroll, 1987). This approach can be integrated with the general view of learning outlined above through the *chunking hypothesis,* and three assumptions which support it (after Rosenbloom and Newell, 1987):

The Chunking Hypothesis: A human acquires and organizes knowledge of the environment by forming and storing expressions, called chunks, which are structured collections of the chunks existing at the time of learning.

Performance Assumption: The performance program of any system is coded in terms of high level chunks, with the time to process a chunk being less than the time to process its constituent chunks.

Learning Assumption: Chunks are learned at a constant rate on average from the relevant patterns of stimuli and responses that occur in the specific environments experienced.

Task Structure Assumption: The probability of recurrence of an environment pattern decreases as the pattern size increases.

This hypothesis suggests that the component chunks of command sequences are progressively clumped together into chunks until a whole command sequence is represented as a single chunk. The fewer chunks are required to perform a command sequence, the less time the sequence takes to perform. Consequently, commands learned at an early stage have no or few relevant constituent chunks, and require the development of a procedure for the sequence from its minimal parts. The representation for commands learned later may include chunks which were developed for commands already learned. However, they still require the development of some structures which were not previously defined, and the appropriate recruitment of those which are. This process may be easier if the new commands conform to an already learned characterization of commands - hence the often cited guideline for consistency in the user interface, and compatibility with previous computer interfaces, and natural language structures (e.g., Barnard et al, 1981). In contrast to the chunks of performance sequences, such characterizations may take the form of high level rules governing the system command structure and operation which users have abstracted.

The results of the Wilson et al (1990) study showed that there are both users and tasks for which performance remains poor and others for which it improves. The major reductions in time are not due to the speeding up of proceduralized tasks, but rather to the ability of some users to identify local errors, and correct them locally, rather than having to go back to the start of a major sequence and re-attempt it completely. It was clear, from verbalizable knowledge, that users had developed general characterizations of the command structure, and indeed they often overgeneralized these rules to apply to the few exceptional commands where they did not, which accounted for a large proportion of the performance errors (e.g., they asserted the need to select menu items for all actions, including the default action of typing into a text editor when no selection was required, and they asserted the need to terminate all menu sequences with a ‘done’ item, which was not always the case).

The individual differences between subjects on this study show the same effects as those in the Halasz and Moran (1983) study summarized above. Subjects who developed clear verbalizable rules about the system, which included the exceptions, performed best on most tasks since they improve their error recovery times. This was the dominant change observed during learning in this study. It is reasonable to classify the users who systematically derived generalization rules and consequently performed better in these tests as employing a *systematic* rather than a *heuristic* learning style (Bariff and Lusk, 1977). Those who performed worse and did not abstract rules may be characterized as employing a *heuristic* learning style, although in this study no explicit measuring was performed of the users learning styles.

A general result supported in the studies above, is that the easiest task action mappings to learn are those which are consistent, compatible, interactive and meaningful. A task action mapping is consistent if it shares structural properties such as syntax with other task-action mappings at the interface. It is compatible if it shares properties with other task action mappings from previously experienced interfaces, or natural language. It is interactive if the task environment perceptually cues the actions which can be performed upon it, and the correct mappings (in the language of Gibson (1979), if the object has an *affordance* for the action). Finally, it is meaningful if the actions can be generated from the semantics of the task, or, as in the previously described studies, if the users can extract generalization rules about the interface to facilitate understanding rather than merely learning by rote.

The Chunking Hypothesis and the associated view of learning have been used as one of the main components in the design of the SOAR problem-solving architecture. SOAR is a problem space theory of human cognitive architecture proposed by Laird, Newell and Rosenbloom (1987) as an integrative theory. In contrast to the specific experimental studies and limited theories described so far, SOAR has been used by Howes and Young (1996) to construct an integrated model of learning and performance at the user interface (TAL).

TAL models a user who starts with some knowledge of the primitive interface actions, for example how to use a keyboard and a mouse, but without knowledge of how to combine these actions into sequences for achieving new goals. Given a task, TAL interacts with a device simulation and an instructor. If it has a rule that determines actions to be performed , those actions are performed. Otherwise, the system requests instructions from the human instructor. The instructor responds by giving some instruction which TAL interprets, and uses to determine the next action to be performed. In this process TAL learns chunks, that encode an interpretation of the instructions. Chunks are rules that consist of a left hand side of conditions and a right hand side of actions. Chunks learned in one task may be transferred to many similar tasks. The more similar the task semantics, the greater the possibility of transferring rules. For example, a chunk from a task to open a file whose condition names a file, and whose action is to move to the file menu, could transfer to all other tasks on that file including closing or saving it,, and to creating a new file. A significant effect of transfer in TAL's behavior is that if transfer is successful then less instruction is required to learn a new set of task-action mappings.

TAL does not attempt to be the complete and correct model whose existence Landauer denied (see section 2.2). Indeed, it is still only in its infancy. It has been used to simulate interaction with both display- and keyboard-based user interfaces, and has produced similar outcomes with many empirical results with respect to consistency, compatibility, interactivity and meaningfulness. It does not contain a model of reportability, so the observed differences between verbalizable and non-verbalizable procedural knowledge can not be captured. Neither does it model error recovery, so many of the phenomena reported above are not captured. Equally, it does not yet implement alternative learning strategies, so the individual differences in learning cannot be duplicated. However, Hegerty, Just and Morrison (1988) have produced a SOAR simulation to account for the results of a study about individual differences in mechanical ability, which could be incorporated into TAL since it has been carried out within the same integrative theory - SOAR.

A widely used test of mechanical ability is the Bennett Mechanical Comprehension Test (Bennett, 1969), on which a large body of data has been collected for the population as a whole, and its predictive power and the variations in scores across the population are well documented. Such psychometric tests allow a quick paper and pencil assessment of a subject, in order to predict their mechanical ability through correlation with previous scores, with a known reliability for a wide range of mechanical tasks. However, there is no explicit or implicit attempt to explain the cognitive processes behind such scores; the tests are a tool designed to serve a specific function. The Bennett test contains items pertaining to many aspects of mechanics including fluid and thermal dynamics, levers, gears and pulleys. Hegerty, Just and Morrison (1988) studied subjects who undertook the pulley questions from the Bennett test, and proposed an explanation and SOAR simulation of the cognitive process that contribute to mechanical ability. The test questions are mostly of the form where two pulley systems are presented, and the subject must decide which one requires more force to lift a weight. Some pulley systems differed only as for a single attribute, while others differed as for more than one attribute, allowing a measure of how subjects combine information from different attributes. Similar problems, in which the variation of both relevant and irrelevant attributes allowed a determination of whether subjects could distinguish the attributes relevant to the mechanical function. A subset of the 43 subjects taking the test gave verbal protocols of the task which were used to derive the rules used by the subjects to answer the test questions, and the preference orderings for the use of rules when they yield different results. A second experiment investigated the range of ways in which subjects addressed problems where both relevant and irrelevant attributes of a problem varied. The range of individual differences was characterized by low scoring subjects using rules based on visible components of the pulley systems. These rules are qualitative, the attributes on which they are based can either be relevant or irrelevant, and subjects have no clear preference among rules, so that their responses appear inconsistent with any particular rule. High scoring subjects, on the other hand, have rules that are quantitative and take configurational properties of the system into account. They prefer rules based on attributes which are highly correlated with mechanical advantage. Three abilities accounted for the individual differences in performance: (1) the ability to discriminate relevant from irrelevant attributes, (2) the consistency of rule use, and (3) the ability to quantitatively combine information about two attributes within a single rule. A simulation model was used to specify mechanisms that can account for these three sources of individual differences. The model suggested that the process of applying rules is similar for high-scoring and low-scoring subjects, but the content of the rules changes with increases in mechanical ability. A subsequent study (Hegarty and Just, 1993) tested eye movements and comprehension of descriptions of pulley systems involving text and diagrams. The results of this study were consistent with the view that the reader's choice of modality depends on the cognitive effort involved in each modality. The construction of mental models from the text emerged as being the less effortful choice for high ability subjects, while the construction of mental models from a diagram as being the less effortful choice for low ability subjects (modality variations will be discussed in more detail in the next section of this Chapter).

These results show that high scoring subjects are flexible problem solvers who can use either qualitative or quantitative mental models, depending on the demands of the problem. For example, some subjects did not resort to using a quantitative model unless the problem required it. If the relative effort required in order to answer the test question could be determined by comparing a single attribute, then just that attribute was evaluated and compared. This invocation of quantitative reasoning when qualitative reasoning is insufficient is generally consistent with the different between expert performance on a task and the performance of novices who only apply qualitative reasoning (De Kleer, 1985; diSessa, 1983).

As TAL develops, there is no reason why reportability effects, error recovery, individual differences and many other phenomena cannot be accounted for. TAL is not the only model of the acquisition of task-action mappings. CE+ (Polson and Lewis, 1990) models the exploratory acquisition of task action mappings. In this system, when the available knowledge is insufficient to achieve a goal, problem solving is performed. Similar approaches have been developed to capture compatibility of designs, the most notable being

Display-based Task-Action Grammar (D-TAG, Howes and Payne, 1990) which has its origins in the competence grammars of linguistics. However, TAL is computationally implemented, whereas, the components of CE+ although implemented in isolation, have not been integrated into a single running simulation. Perhaps, most importantly, TAL implements a single learning system in SOAR's chunking, which operates in both plan-based tasks and those tasks where the cognition is strongly situated in the environment, due to the perceptual cues or *affordances* it offers.

These examples of experimental studies, localized theories, and the use of an integrative theory to host different models, shows how the elements provided by cognitive science can come together. However, these results are not yet of direct practical use in designing user interfaces for all, since they require cognitive science skills to both apply and interpret them. Despite this, they provide an understanding of how the learning process is modeled, and can move on to address individual differences. This, in turn, provides a basis for user interfaces that *support all potential users throughout system learning and use.*

3.3 Modality modelling and design choices

SOAR provides an integrative theory within which models of experimental data derived from different paradigms can be constructed. However, SOAR primarily represents information at a propositional level, which is appropriate for reasoning or planning tasks, but is less compatible with the problems and variations encountered in more perceptual tasks involving sensory modalities (e.g. voice, vision), and the performance of action with a range of devices. Such modality issues must be modeled in order to design interfaces for those with accessibility problems due of sensory or action impairments, to overcome interaction problems which may only arise for users without impairments in rapidly changing environments (such as flying planes, or managing nuclear reactors), and to investigate new interface devices which may support novel interaction methods (such as virtual or augmented reality).

Interacting Cognitive Subsystems (ICS) represents the human information processing system as a highly parallel organization with a modular structure. The ICS architecture (Barnard, 1985) contains a set of functionally distinct subsystems, each with equivalent capabilities, yet each specialized to deal with a different class of representation. The ICS architecture can be considered as an integrative framework (in contrast to SOAR which is an integrative theory) in which some components are further specified as theories, or indeed models, while others are outlined only at the framework level, awaiting elaboration from other theories.

The ICS subsystems exchange representations of information directly, with no role for a *central processor* or *limited capacity working memory* <<<PLEASE CLARIFY>>>. Acting together, nine component subsystems deal with incoming sensory information, structural regularities in that information, the meanings that can be abstracted from it, and the creation of instructions for the body to respond and act both externally, in the real world, and internally, in terms of physiological effects. Figure 1 outlines the overall architecture, while Table 1 lists the nature of the mental representations which each subsystem processes. The subsystems are classed as peripheral if they exchange information with the world via the senses of the body, and as central if they only exchange information with other subsystems.

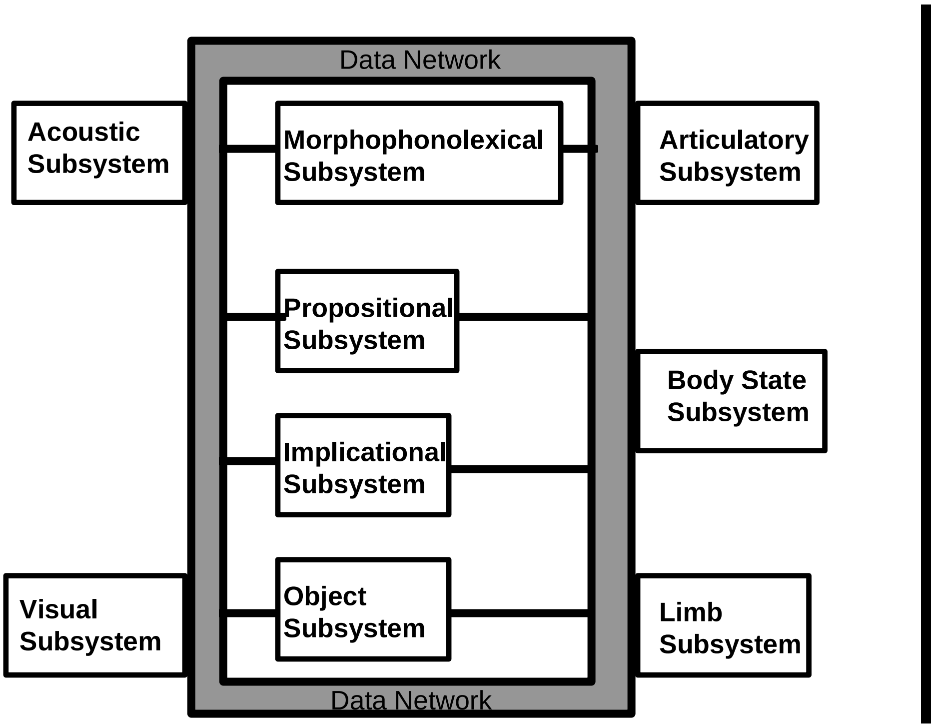

Figure 1: The systematic organization of ICS.

Each subsystem has the same internal structure, even though they address different classes of information. They each receive a representation at an input array which is *copied* to an Image Record, while simultaneously being operated on by a number of *transformation processes*. The Image Record acts as a local memory for the subsystem. Any representation ever received is stored in the Image Record, and in the long term, any commonalties and regularities of the representations in a subsystem can be abstracted from the stock of past experience. The *transformation processes* are the key to the function of the overall organization. In normal operation, these processes transform the information represented on the input array into a different representation, for use by another subsystem. For example, the VIS subsystem contains a VIS-> OBJ transformation process that transforms the visual information into a more abstract object representation. These transformation processes are independent and work in parallel, with the consequence that a subsystem can produce multiple simultaneous outputs.

Subsystems do not contain transformation processes from their own to all other representations, with the consequence that chains of subsystem operation may be required to produce the final representation. For example, to transform information from a visual to a propositional representation requires the intermediate use of the object subsystem, since a direct transformation is not available. A consequence of this is that cognition is a result of a series of subsystems acting in a chain or configuration as information flows through the system. The more complex configurations can include cyclical exchanges of information between pairs and even triplets of central subsystems (e.g. PROP-> IMPLIC & IMPLIC -> PROP, and OBJ-> MPL, MPL-> PROP & PROP->OBJ). Having built up an understanding of the resources needed to perform a particular task using a particular interface, it is then possible to reason about the suitability of the interface by, for example, looking at the conflicts that arise in the use of the identified cognitive resources. ICS provides a framework for answering such questions as “how many information channels can be used simultaneously?” using concepts such as stability of configurations, oscillation between configurations competing for resources, and the requirements of blending different data streams.

PERIPHERAL SUBSYSTEMS	
a) Sensory	
1) Acoustic (AC)	Sound frequency (pitch), timbre intensity etc. Subjectively ‘what we hear in the world’.
2) Visual (VIS)	Light wavelength (hue), brightness over visual space etc. Subjectively ‘what we see in the world’ as patterns, shapes and colors.
3) Body State (BS)	Type of stimulation (e.g. cutaneous pressure, temperature, olfactory muscle tension) its location, intensity etc. Subjectively, bodily sensations of pressure, pain, positions of parts of the body, as well as tastes and smells etc.
b) Effector	
4) Articulatory (ART)	Force, target positions and timing of articulatory musculature (e.g. place of articulation). Subjectively, our experience of subvocal speech output.
5) LIMB (LIM)	Force, target positions and timing of skeletal muscles Subjectively, ‘mental’ physical movement.
CENTRAL SUBSYSTEMS	
c) Structural	
6) Morphophonolexical (MPL)	An abstract structural description of entities and relationships in sound space. Dominated by speech forms, where it conveys a surface structure description of the identity of words, their status, order and the form of boundaries between them. Subjectively, what we ‘hear in the head’, our ‘mental voice’.
7) Object	An abstract structural description of entities and relationships in visual space, conveying the attributes and identity of structurally integrated visual objects, their relative positions an dynamic characteristics. Subjectively, our ‘visual imagery’.
d) Meaning	
8) Propositional	A description of entities and relationships in semantic space conveying attributes and identities of underlying referents and the nature of the relationships among them. Subjectively, specific semantic relationships - ‘knowing that’.
9) Implicational (IMPLIC)	An abstract description of human existential space, abstracted over both sensory and propositional input, and conveying ideational and affective content: schematic models of experience. Subjectively, ‘senses’ of knowing (e.g. ‘familiarity’) or of affect (e.g. apprehension, desire).

Table 1: The subsystems within ICS and the type of information with which they deal.

As TAL and CE+ represent both the user and the system, so ICS can be represented in a formal notation along with a system. Modal Action Logic (MAL) has been used on several occasions to reason about gestural interaction, or about interaction using novel input and display devices (Duke, 1995; Duke et al, 1995) in an approach termed *syndetic modeling* (Faconti and Duke, 1996). Such a formal modeling approach requires a detailed understanding of the notations used, and take considerable effort to perform. Clearly, this is not required in every interface design. However, to understand the properties of novel interface devices, which may provide support for users with individual needs due to some impairment in one or more sensory modality, they can be efficient as well as effective in pursuing the goal of *User Interfaces for All*.

4. UI development tools and methods as cognitive science in practice

The previous section has described an integrative theory (SOAR) instantiated with a model (TAL) that draws on experimental data collected from various paradigms to model learning, and an integrative framework (ICS) which can be used in conjunction with formal system models (as syndetic modeling) to model device level interactions. These approaches have potential to provide insights into HCI, which could contribute to the development of *User Interfaces for All*, but these approaches also require considerable craft skill in cognitive science to use them; in both cases the techniques are models of the user's task performance *per se*, and they need models to be populated to a fine granularity, thus requiring considerable effort.

There are very few user interface development tools or methods, which have been derived from cognitive science, that do not suffer from these shortcomings. Available tools model the task to be performed using constructs derived from human information processing models, rather than presenting models of the human processing explicitly. They also assume the simplest cases of static knowledge of an expert, rather than addressing the dynamic knowledge of learning. However, these constraints allow them to be specified without cognitive science skill, and to be specified at either coarse or fine granularities, making them adequate for use by system developers when the benefits of modeling outweigh the costs (Bias & Mayhew, 1994).

Two methods are briefly described here (GOMS and KADS), illustrating how they capture a range of tasks - *User Interfaces for All Tasks* - and how they capture the variation in the performance of those tasks by users as selection rules to choose between methods, or as alternative strategies. GOMS models a task as a set of Goals, Operators, Methods and Selection rules for UI design evaluation, while KADS models reasoning tasks in terms of domain knowledge, inference rules, task structures, and strategies for requirements engineering.

4.1 GOMS - Engineering Task Performance

The GOMS model was developed by Card et al (1983) based on previous work in the domain of human problem solving by Newell and Simon (1972). GOMS has been designed as a cognitive engineering model. Therefore, its primary purpose is to be used in the design of user interfaces rather than as a foundation of scientific theories. The model is used to outline the cognitive performance of a person by decomposing a problem into hierarchical goals and goal stacks. The basic GOMS model is best at making qualitative predictions about differences between tasks in which users make no or few errors. By associating times, or time distributions with each operator, GOMS models are able to make total performance time or statistical predictions. Depending on the granularity of analysis, several variations of GOMS models can be explored to make quantitative predictions, e.g. from unit-task to keystroke level.

The four basic elements of a GOMS model are Goals, Operators, Methods and Selection Rules. A goal is a symbolic representation of a state of affairs to be achieved, which determines a set of possible methods to be used to achieve it. Operators are elementary perceptual, motor, or cognitive acts whose execution is necessary to change any aspect of the user’s strategy or the task environment. Methods describe procedures for achieving goals in terms of operators, or other goals. Operators that are often used together are grouped into methods, in the way that the chunking mechanism in SOAR would. However, GOMS assumes all such expert chunks have been constructed, and includes no learning mechanism, thus being applicable only to expert performance. The fourth elements are rules to select between alternative methods to perform goals –(Selection Rules).

GOMS has received much more empirical testing than any other analytic model of human-computer interaction tasks (Gugerty, 1993). Models have been developed for a wide variety of applications including simple text editors (Card et al, 1983), spreadsheets (Olsen and Nilsen, 1988), and hypertext applications (Carmel et al, 1992). The strongest validation of its use has been on an interface for telephone inquiry operators where cutting task time by seconds mounts up to savings of millions of dollars over a year (Gray et al, 1990, 1993). However, this is primarily a sensorimotor task performed by highly skilled operators, which is where the model is intended to perform best.

GOMS analysis allow flexibility for modeling variations of task performance with respect to the times chosen for operators, which will only effect the quantitative predictions of performance time, and in the Selection Rules. Selection Rules provide the control structure to the model in terms of if-then rules. In an example where different methods are available for cursor control, a selection rule may state:

> *IF the desired position and the current position are both on the screen THEN the arrow key method would be used.*
>
> *IF the desired position is on a different screen than the current position*

THEN the search command method would be used.

The original presentation of GOMS (Card, et al, 1983) stated that expert method selection would always be made on the basis of the fastest method, and this reasoning should be used in constructing selection rules. Although this is a useful guideline for developing engineering models, it has been empirically shown to be false (Maclean et al, 1985), and users show some inertia against changing methods from one, recently used, to another, even if the second is faster. Clearly, the data on learning, reported above, , and the *training wheels* approach to learning computer systems, suggest that for non-experts, many methods may not be within the habitable subset of commands, and may be selected only very rarely, even when they are considerable more efficient.

In order to make the GOMS approach more usable for practical applications, Kieras (1988) developed the Natural GOMS Language (NGOMSL), which allows the modeler to describe user computer interaction in a specification language similar to computer programming languages. Cognitive Complexity Theory (CCT) represents another extension of the basic GOMS model (Kieras and Polson, 1985). With respect to CCT, the CE+ proposal mentioned above (Polson and Lewis, 1990) is a further advance to address ‘walk up and use’ interfaces (e.g. public information kiosks). John (1988) has extended the approach to the analysis of parallel activities. The GOMS approach and its followers have dominated the research on models in HCI. However, to date, such an approach has had little influence on the practice of UI design. The technique is still very dependent on the skill of the analyst, and although Kieras’ work has addressed learning time predictions, expert-novice differences, and estimates of mental workload from the number of items in working memory at any time, the approach is not tuned to address individual differences (Olsen and Olsen, 1990).

4.2 KADS - Engineering Task Specification

GOMS provides a tool to predict interaction time, and some aspects of complexity for a clearly specified task, once a system has been designed. Cognitive science has also produced task models which can guide the developer at an earlier stage of system development, when a problem is being analyzed. . KADS was developed as a Knowledge Acquisition and Design System for knowledge based systems with a firm theoretical foundation (see Schreiber et al, 1993), as well as a clearly stated method to be followed by developers (see Tansey and Hayball, 1993). KADS provides a library of task models which can be instantiated to become descriptions of future systems and their domains.

As a complete method, KADS provides models of: the system; the context within which it will work; the users; cooperation with other agents; the system’s organizational environment; and, the details of the documentation required for the complete development process from analysis through design to implementation (see Tansey and Hayball, 1993). The current description will only cover the central library of expertise models which can be applied widely to guide requirements engineering.

The problem solving components of tasks is investigated and described by building up an Expertise Model from its four layers: the Domain Layer which describes the static factual knowledge about the domain; the Inference Layer which defines the inference steps which the system can make; the Task Layer which defines the basic problem solving tasks; and the Strategy Layer which defines how tasks are constructed, modified or chosen. Although an instantiated expertise model is an implementation independent representation of the problem-solving capability of a prospective system, it is more concrete than an uninstantiated model would be. KADS provides a library of uninstantiated models for the inference and task layer called Generic Task Models, which can be used to motivate the analysis (see Breuker and Van de Velde, 1994). Each of the four layers of the Expertise Model will be outlined here.

The Domain Layer defines static domain knowledge as ontology, consisting of structures of domain concepts and relationships between concepts. This knowledge is termed 'static' since it is describes a domain while being neutral as to how it is used for inference purposes.

The Inference Layer describes basic inferential capability, in terms of inference types, domain roles and inference structures linking these with tasks at the task layer. It identifies the inferences that can be made for selected tasks over the static knowledge in the domain layer. 'Inference types' are descriptions of the way in which domain concepts, structures or relations can be used to make inferences. For example, the 'classify' inference type takes an object with its attributes, and derives the class of the object. Inference types direct the way in which static domain knowledge may be used, and provide 'handles' for the control of inference by the next, Task Layer. 'Domain roles' define the functions which domain structures may perform. For example, in a diagnosis task in the medical domain, HIV may be either a hypothesis to be verified or a solution resulting from a reasoning process. These are two different roles for a single domain concept within a problem solving process. The third component in the layer is the 'inference structure', which is a network of inference types and domain roles constraining a reasoning process by explicitly describing which inferences can be made, and implicitly defining which cannot be made. Inference structures are depicted in a graphical form where domain roles are represented by rectangles and inference types by ellipses (for example, Figure 2).

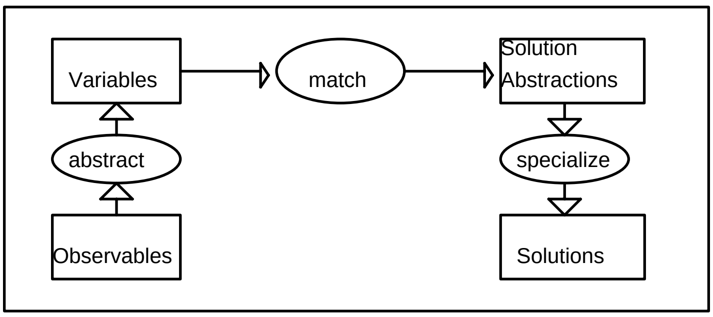

Figure 2: Inference Structure for Heuristic Classification

The third layer in the Expertise Model is the Task Layer, which describes how the individual inferences within the Inference Layer may be sequenced in order to satisfy the required problem solving goals. The representation used is a task structure which can be defined statically (with a fixed control structure) or dynamically (e.g. as a result of planning at the inference layer). Task Structures are typically simple sequences of inferences wrapped in some conventional procedural control structures, such as selection (IF ... THEN... ELSE) and repetition (e.g. FOR, WHILE and REPEAT). However, they may include more complex control structures such as parallelism, and temporal dependency.

The fourth layer of the Expertise Model is the Strategy Layer, which provides the strategic knowledge to select, sequence, plan or repair the corresponding Task Structures. The following types of strategic knowledge may be described: Goal selection, Task structure selection, Goal sequencing, Task structure configuration, mode of system operation, inference control and repair.

KADS allows analysts to approach problem solving tasks with poorly expressed algorithms in the same way as data intensive tasks or detailed algorithm implementation may be approached. However, the major contribution of KADS addresses the identification of problem solving tasks and the re-use of task level descriptions through Generic Task Models. KADS provides a library of Generic Task Models (GTMs) which are models of problem solving tasks e not tied to a particular domain. GTMs are used to initiate and drive the analysis process during Expertise Model development. For each task, the model describes it at the Task and Inference Layers. Both the Domain and Strategic layers are strongly domain dependent and therefore little generic information can be given for them.

- SYSTEM ANALYSIS
    - Identification
        - Diagnosis
            - Single Model Diagnosis
                - **Systematic Diagnosis**
                    - **Localization**
                    - **Causal Tracing**
            - Multiple Model Diagnosis
                - **Mixed Mode Diagnosis**
        - **Verification**
        - **Correlation**
            - **Assessment**
        - **Monitoring**
        - Classification
            - **Simple Classification**
            - **Heuristic Classification**
            - **Systematic Refinement**
    - **Prediction**
        - **Predication of Behavior**
        - **Prediction of Values**

• SYSTEM MODIFICATION
- Repair
- Remedy
- Control
- Maintenance

• SYSTEM SYNTHESIS
- **Design**
  - **Hierarchical Design**
  - **Incremental Design**
- Configuration
  - **Simple Configuration**
  - **Incremental Configuration**
- **Planning**
- **Scheduling**
- **Modeling**

Table 2: The hierarchy of Generic Task Models in the KADS Task Library

The library of GTMs is presented as a hierarchy. The top node represents tasks, the next layer has three entries: (i) Systems Analysis, which deal with an examination of the elements or structure of some entity; (ii) System Modification which deals with tasks which update or change some entity (often after a process of analysis and synthesis, but which modify the entity after finding a solution); and (iii), System Synthesis, which deals with tasks which build up an entity from constituent parts. The full hierarchy of GTMs is presented in Table 2.

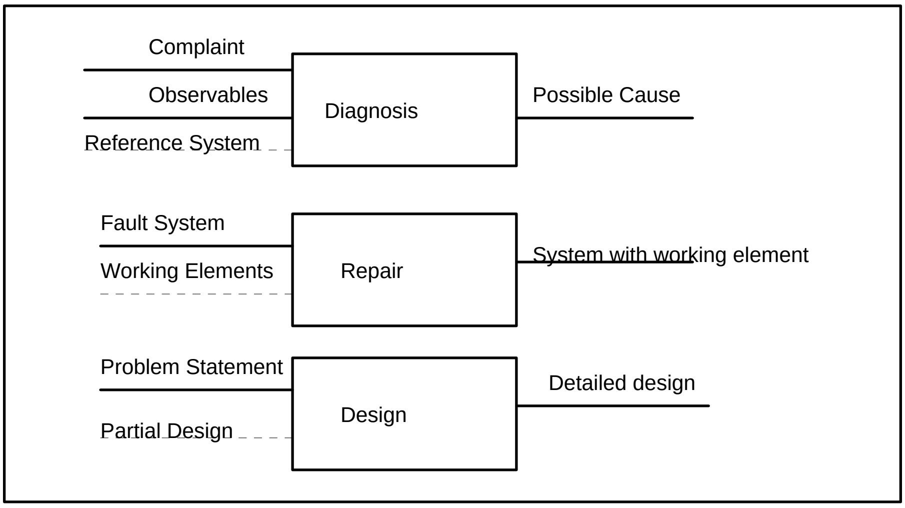

Figure 3: Input / Output descriptions of three task models (input on left, output on right; solid lines necessary, dashed lines optional).

For each GTM, the input and output descriptions are presented (see Figure 3) to aid the analyst in selecting the appropriate GTM for a system by considering the knowledge based tasks in the Process Model in terms of their inputs and outputs. GTM's are used as a starting point for developing Expertise Models. The selection of the GTM and the Analysis of Static Knowledge are the first two stages of the expertise analysis, and are performed in parallel. Once the GTM is selected, the specialization of its terminology to that of the problem domain collected in the Analysis of Static Knowledge can begin. This population is then continued as the model develops through step-by-step refinement, eliciting knowledge from documents, task simulations and through structured interviews.

Both GOMS and KADS allow tasks to be modeled at different granularities. In the case of GOMS, at the unit task level, or lower levels down to the keystroke. In the case of KADS, at the task model level, or lower levels down to complete domain knowledge ontologies. Equally, both GOMS and KADS offer a mechanism to choose between user strategies, through selection rules in GOMS, or strategic knowledge rules in KADS. Both are derived from cognitive science models of human information processing, but are abstracted away from the processing itself, and provide representation languages, or libraries embodying them. In both cases the methods are accessible to system developers for use in designing systems for a range of tasks and users.

5. Conclusion

This chapter has tried to convey the spirit of the cognitive science enterprise , and to show how HCI can benefit from the results of freedom in modeling behavior by using cognitive representations and processes . The use of integrative theories and frameworks to capture local theories and experimental results of cognitive science has been illustrated with TAL and ICS. The limitations in the accessibility of these theories and frameworks to system developers are overcome in methods which hide the underlying cognitive processing, and model task representations, as illustrated by GOMS and KADS. However, these accessible tools are limited in the range of phenomena they can address, as a result of their simplifying assumptions.

Variations in the way different users learn, and the different knowledge and skills that they start with, must be addressed within the *User Interfaces for All* approach. A major problem with developing tools based on cognitive science theory is that it requires detailed descriptions of knowledge, which are usually created too late to influence the technology because of the need for experimental data from those technologies once they are created to tune models - “Human computer interaction … will not sustain approaches that are too low level, too limited in scope, too late and too difficult to apply in real design …” (Carroll and Campbell, 1986). This paradox has no clear solution, although cognitive science clearly has much to offer the *User Interfaces for All* approach.

Perhaps the greatest influence of cognitive science on user interfaces, and on the *User Interfaces for All* approach in the future, has, paradoxically, not been mentioned here, but in other chapters in this volume. The cognitive science method produces models and representations of mental life which are normally implemented on computers. The human is the only generally accepted example of intelligent mental life we have. Therefore these models are the best detailed models of intelligent mental life we have which are amenable to computer programming. Cognitive science models of human reasoning with prototypes were used as the basis of most of the early work on user modeling described in chapter 20 by Kay. Cognitive science models of human communication comprehension and generation were the basis of the architectures of intelligent multimedia systems described in chapter 4 by Maybury.

The above mentioned systems, which adopt cognitive science models as the inspiration for representations and architectures, have already been developed. If the cognitive science method is too detailed and slow to refine interface design once innovation has happened, the important question to answer is - *what will be the next innovation, and what does cognitive science offer to guide it?*

The 1998 user interface, like the 1984 one, is a two dimensional graphical representation of the desktop controlled by direct manipulation. Recent advances in several core software technologies have made possible a new type of human-computer interface: the conversational character. Conversational characters are autonomous, anthropomorphic, animated figures (Badler et al, 1993) that have the ability to communicate through multiple modalities, including spoken language, facial expressions (Sproull et al, 1996; Pelachaud et al, 1996), and gestures. Conversational characters inhabit three-dimensional virtual worlds along with avatars of humans, to facilitate human-human (Benford et al, 1995), as well as human-computer, interaction. Unlike textual natural language interfaces, conversational characters have the ability to perceive and produce the verbal and non-verbal signals that identify discourse structure and regulate the flow of information between interlocutors. Such signals include intonational patterns, gestures, back-channel feedback signals, and turn-taking protocols. These capabilities enable them to engage in complex interactions with human users via natural speech, rather than complex command languages, menus or graphical manipulations. For conversational characters to maintain realistic embodiments, they require rich representations of emotions, goals, plans, affordances, learning, and cultural and other individual differences, which must be adequately conveyed in conversation in order to maintain a sense of presence and engagement.

The fully anthropomorphic interface envisioned in Arthur C. Clark’s 2001 as HAL 9000 is still a long way off (see Stork, 1996). However, cognitive science has many models that provide insights into how to construct such complex "beasts". Much of the research in the last ten years in cognitive science has been directed at integrating emotional aspects of mental life with the cognitive aspects (Ortony et al, 1988; Oatley and Johnson-Laird, 1987). A second advance has been to address discourse and dialogue rather than the syntax. If you agree that conversational characters in three dimensional multi-user virtual worlds are the way of the future, and this chapter has helped you understood the basis of cognitive science models, then cognitive science research may be the place to look further for the insights which will provide *User Interfaces for All.*

Acknowledgments

The research reported in this paper was partly funded by CEC Esprit IV grant 20597 to the CHAMELEON project, partly by CEC HCM grant CHRX-CT93-0085 to the ERCIM

Computer Graphics Network, and partly by CEC TMR grant FMRX-CT97-01333 to the TACIT consortium.

References

Anderson, J.R. (1983). *The Architecture of Cognition*. Mass.: Harvard University Press.

Badler, N.I., Phillips, C.B., and Webber, B.L. (1993*) Simulating Humans: computer graphics animation and control.* Oxford University Press, 1993.

Bariff, M.L. and Lusk, E.J. (1977) Cognitive and personality tests for the design of management information systems. *Management Science*, **23**, 820-829.

Barnard, P.J. (1985) Interacting Cognitive Sybsystems: A psycholinguistic approach to short term memory. In A. Ellis (ed.) Progress in the Psychology of Language, (Vol 2). London : Lawrence Earlbaum Associates. 197-258.

Barnard, P.J. (1995) The contributions of applied cognitive psychology to the study of human computer interaction. In R. Becker, J. Grudin, W. Buxton, S Greenberg (Eds) *Readings in Human Computer Interaction: Towards the Year 2000,* San Francisco: Morgan Kaufmann. 640-658.

Barnard, P.J., Hammond, N., Morton, J., Long, J and Clark, I. (1981) Consistency and Compatibility in human computer dialogue*. International Journal of Man Machine Studies*, **15**, 87-134.

Benford, S., Bowers, J., Fahlen, L.E., Greenhalgh, C. and Snowdon, D. (1995) User Embodiment in Collaborative Virtual Environments. In *Proceedings of CHI '95*, 242-249. New York: ACM Press.

Bennett, C.K. (1969) *Bennett Mechanical Comprehension Test.* New York: The Psychological Corporation.

Bias, R.G, and Mayhew, D.J. (1994) *Cost-Justifying Usability*, London: Academic Press.

Breuker, J. And Van de Velde, W (1994) *CommonKADS Library for Expertise Modelling*. Amsterdam: IOS Press.

Card, S.K., Moran, T.P. and Newell, A. (1983*) The Psychology of human-computer interaction*. Hillsdale: Lawrence Earlbaum Assoc.

Carmel, E., Crawford, S., and Chen, H. (1992) Browsing in Hypertext: a cognitive study. *IEEE Transactions on Systems, Man and Cybernetics*, **5**, 865-884.

Catrambone, R. And Carroll, J.M. (1987) Learning a word processor system with training wheels and guided exploration. In J.M. Carroll and P.P. Tanner (eds*) Proceedings of CHI + GI 87*, New York, ACM.

Carroll, J.M. and Campbell, R.L. (1986) Softening up hard science: reply to Newell and Card. *Human Computer Interaction*, 2(3) 227-250.

De Kleer, J. (1985) How Circuits Work. In D. Bobrow (ed*.) Qualitative reasoning about physical systems*. Cambridge, MA: MIT Press.

DiSessa, A.A. (1983) Phenomenology and the evolution of intuition. In D. Gentner & A.L. Stevens (eds) *Mental Models*. Hillsdale, NJ: Erlbaum.

Duke, D.J. (1995) Reasoning about gestural interaction. *Computer Graphics Forum*, **14**(3); C-55-C-66.

Duke, D.J., Barnard, P.J., Duce, D.A., and May, J. (1995) Systematic development of the human interface. In *Proceedings of the Second Asia-pacific Software Engineering Conference APSEC '95*, 313-321. IEEE Computer Society Press.

Ericsson and Simon (1984) *Protocol Analysis: Verbal Reports as Data*. MIT Press: Cambridge, Mass.

Faconti, G.P. and Duke, D.J. (1996) Device Models. In F. Bodart and J. Vanderdonckt (eds) *Design, Specification and Verification of Interactive Systems '96*. Vienna: Springer-Verlag.

Fitts, P.M. (1964). Perceptual-motor skill learning. In A.W. Melton (ed.) *Categories of Human Learning*. New York: Academic Press.

Furnas, G.W., Gomez, L.M., Landauer, T.K. and Dumais, S.M. (1982) Statistical semantics: how can a computer use what people name things to guess what people mean when they name things*? Proceedings of Human Factors in Computer Systems*. New York: ACM. 251-253.

Furnas, G.W., Gomez, L.M., Landauer, T.K. and Dumais, S.M. (1983) Statistical semantics: anlysis of the potential performance of key-word information systems, *The Bell System Technical Journal*, 1753-1906.

Gibson, J.J. (1979) *The ecological approach to visual perception*. Boston: Houghton Mufflin.

Gomez, L.M.and Lockbaum, (1984) People can retrieve more objects with enriched keyword vocabularies. But is there a performance cost ? In B. Shackel (ed.) *Human Computer Interaction - Interact 84*. Amsterdam: North Holland, 257-261.

Gray, W, John, B., Stuart, R., Lawrence, D. and Attwood, M. (1990) GOMS meets the phone company: Analytic modelling applied to real world problems. *Proceedings INTERACT '90*. Amsterdam: Elsevier, 29-34.

Gray, W, John, B. and Attwood, M. (1993) Project Ernestine: A validation of GOMS for prediction and explanation of real-world task performance. *Human Computer Interaction*, **8**(3), 237-309.

Gugerty, L. (1993) The use of analytic models in human-computer interface design. *International Journal of Man-Machine Studies*, **38**, 625-660.

Halasz, F. and Moran, T.P. (1983) Mental Models and problem solving using a calculator. *Proceedings of CHI '83,* (Boston), 212-216, New York: ACM.

Hegarty, M. And Just, M.A. (1993) Constructing Mental Models from Text and Diagrams, *Journal of Memory and Language*, **32**, 717-742.

Hegarty, M., Just, M.A. and Morrison, I.R. (1988) Mental Models of Mechanical Systems: Individual Differences in Quantitative and Qualitative Reasoning, *Cognitive Psychology*, **20**, 191-236.

Howes and Payne, (1990) Display-based competence: Towards user models for display driven interfaces. *International Journal of Man Machine Studies*, **33**, 637-655.

Howes, A. and Young, R.M. (1996) Learning Consistent, Interactive, and meaningful Task-Action Mappings: A computational model. *Cognitive Science,* **20**(3) 301-356.

John, B. (1988) *Contributions to engineering models of human computer interaction*. Unpublished doctoral dissertation, Carnegie Mellon University, Pittsburgh, PA.

Kieras, D.E. (1988) Towards a practical GOMS model methodology for user interface design. In M. Helander (ed.) *Handbook of human computer interaction*. Amsterdam: North Holland, 135-157.

Kieras, D.E. and Polson, P.G. (1985) An approach to the formal analysis of user complexity. *International Journal of Man Machine Studies*, **22**, 365-394.

Laird, Newell and Rosenbloom (1987).SOAR: an architecture for general intelligence. *Artificial Intelligence*, **33**, 1-64.

Landauer, T.K. (1987) Relations between cognitive psychology and computer systems design. In J.M. Carroll (ed.*) Interacting Thought: Cognitive Aspects of Human-Computer Interaction.* Cambridge MA: MIT Press, 1-25.

Landauer, T.K. (1996) Let's Get Real: A position paper on the role of cognitive psychology in the design of humanly useful and usable systems. In R. Becker, J. Grudin, W. Buxton, S Greenberg (Eds) *Readings in Human Computer Interaction: Towards the Year 2000,* San Francisco: Morgan Kaufmann.

MacLean, A., Barnard, P.J., and Wilson, M.D. (1985) Evaluating the human interface of a data entry system: user choice and performance measures yield different trade-off functions. In P. Johnson and S. Cook (eds.) *People and Computers: Designing the Interface*. Cambridge: Cambridge University Press, 172-185.

Miller, G.A. (1966) *Psychology: the science of mental life*, London: Pelican Books.

Morton, J., Barnard, P.J., Hammond, N.V., and Long, J. (1979) Interacting with the computer: a framework. In E.J. Boutmy and A. Danthine (eds.) *Teleinformatics '79*. Amsterdam: North Holland, 201-208.

Neisser, U., Novick, R. and Lazar, R. (1963) Searching for ten targets simultaneously. *Perceptual and Motor Skills*, **17**, 427-432.

Neves, D.M. and Anderson, J.R. (1981) Knowledge compilation: Mechanisms for the automatisation of cognitive skills. In J.R. Anderson (ed.) *Cognitive Skills and Their Acquisition*. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Newell, A. (1990) *Unified theories of cognition*. Cambridge MA: Harvard University Press.

Newell, A. and Rosenbloom, P.S. (1981) Mechanisms of skill acquisition and the law of practice. In J.R. Anderson (ed.) *Cognitive Skills and Their Acquisition*. Hillsdale, N.J.: Lawrence Erlbaum Associates.

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

Norman, D.A. (1993) An Introduction to the Special Issue on Situated Action. *Cognitive Science*, **17**(1), 1-7.

Oatley, K. and Johnson-laird, P.N. (1987) Towards a cognitive theory of emotions. *Cognition and Emotion*, **1**, 29-50.

Olsen, J.R. and Nilsen, E. (1988) Analysis of the cognition involved in spreadsheet software interaction. *Human Computer Interaction,* **3**, 309-350.

Olsen, J.R. and Olsen, G.M. (1990) The growth of cognitive modelling in human computer interaction since GOMS. *Human Computer Interaction*, **5**, 221-265.

Ortony, A., Clore, G.L., and Collins, A. (1988) *The cognitive structure of emotions*. New York: Cambridge University Press.

Payne, S. and Green, T. (1986) Task-action grammars. A model of the mental representation of task languages. *Human Computer Interaction*, **2**, 93-133.

Pelachaud, C., Badler, N.I., and Steadman, M. (1996) Generating Spatial Expressions for Speech. *Cognitive Science*, **20**(1), 1-46.

Polson, P.G. and Lewis, C.H. (1990) Theory based design for easily learned interfaces. *Human Computer Interaction*, **5**, 191-220.

Popper, K.R. (1992) *The Logic of Scientific Discovery*. London: Routledge.

Schreiber, G., Wielings, B. & Breuker, J.: *KADS: A principled Approach to Knowledge Based Systems Development*. Academic Press: London, 1993.

Snoddy, G.S. (1926) Learning and Stability. *Journal of Applied Psychology*, **10**, 1-36.

Sproull, L., Subramani, M., Kiesler, S., Walker, J.H. and Waters, K. (1996) When the interface is a face. *Human Computer Interaction*, **11**(2), 97-124.

Sternberg, S. (1969) On the discovery of processin stages: some extensions to Donder's method, *Acta Psychologica*, **30**, 276-315.

Stork, D.G. (1996) *HAL's Legacy: 2001's computer as dream and reality*, MIT Press: Cambridge, Mass. .

Suchman, L. (1987) *Plans and situated action: The problem of human-machine communication.* New York: Cambridge University Press.

Tansley, D.S.W. & Hayball, C.C.: *Knowledge Based Systems Analysis and Design: A KADS Developer's Handbook*. Prentice-Hall: London, 1993.

Wilson, M.D., Barnard, P.J. and MacLean, A. (1990) An investigation of the learning of a computer system. In P. Falzon (ed.) *Cognitive Ergonomics: understanding, learning and designing human computer interaction*. London: Academic Press.

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