CHAPTER 17

Cognitive Models of Human–Information Interaction

Peter Pirolli
Palo Alto Research Center, USA

INTRODUCTION

Human–information interaction (HII) is an emerging branch of human–computer interaction (HCI) which is concerned with how people interact with and process outwardly accessible information such as the World Wide Web.[1] However, HII adopts an information-centric approach rather than the computer-centric approach to the field of human–computer interaction (Lucas 2000). Like HCI, HII is an application field that provides a complex test bed for theories of *cognitive architecture*. In turn, such theories provide the basis for *cognitive engineering models* that can yield predictions about technology and information design. This chapter provides an overview of cognitive architectures and cognitive engineering models in the context of human–information interaction.

The evolution of HCI toward the information-centric field of HII has occurred because of the increasing *pervasiveness* of information services, the increasing *transparency* of user interfaces, the *convergence* of information delivery technologies, and the trend toward *ubiquitous computing* (Lucas 2000). Access to the Internet is pervasive through land lines, satellite, cable, mobile devices, and wireless services. The field of HCI over the past two decades and more has led to the development of computers and computer applications that are increasingly transparent to users performing their tasks. In parallel, the business world around consumer media technologies shows excitement over the convergence of television, cell phones, PCs (personal computers), PDAs (Personal Digital Assistants), cars, set-tops, digital music players, and other consumer electronic devices, as well as the convergence among the means for transporting information, such as the Internet, radio, satellite, cable, and so on. Research on ubiquitous computing looks forward to a world in which computational devices are basically everywhere in our homes, mobile devices, cars, and so on, and these devices can be marshaled to perform arbitrary tasks for users. The net effect of these trends is to make computers invisible, just as electricity

Handbook of Applied Cognition: Second Edition. Edited by Francis Durso.

and electric motors are invisible in homes today (Norman 1998). As computers become invisible and information becomes copious and pervasive, we expect to see a continuing shift in studies from human–computer interaction to human–information interaction. Digital content is becoming independent of the particular physical storage devices and interaction devices. Rather than focus on the structure of devices and application programs, the focus of HII research centers on interaction with content and interactive media.

Although information is part of the focus of the field of HII, it is not the sole focus. HII continues to share with HCI a focus on the psychology of users. Information per se, whether in the classical sense (patterns of organization) or common sense (documents, email, summaries, document clusters, search results, etc.), is of limited interest. Information content has the potential to be used in ways that improve the achievement of human purposes. Information itself is best understood in relation to human use of that information, so human intentionality, psychology, and activity are crucial to providing coherence to the study of human–information interaction.

This chapter takes a particular approach to the psychology of HII, with a focus on the cognitive architectures and cognitive engineering models that are being extended to deal with HII questions. Cognitive engineering in the domains of HCI and HII is founded on the assumption that psychology ought to be able to predict the consequences of different technology designs. For instance, cognitive engineering models have been developed to address questions such as:

- How much time would it take to perform elementary tasks, like inserting, deleting, or moving text?
- How long will it take to learn the skills required for basic text editing?
- Will knowledge of other applications, such as a spreadsheet, transfer to the text editor?
- Will a user be able to figure out how to perform tasks (e.g., by exploration of the interface) without explicit instruction?
- How long will it take an experienced user to find an answer to a question using their PDA?
- What arrangement of information on a display yields more effective visual search?
- How difficult will it be for a user to find information on a website?

For modern information systems, iterative empirical testing and design revision are usually too expensive and too slow. One solution to this practical problem has been the development of *discount usability methods* (e.g., Nielsen & Mack 1974; Spool *et al.* 1999; Nielsen 2000) that employ low-cost techniques such as think-aloud usability tests and heuristic evaluations with small numbers of subjects (often experts in the target domain of interest), often using low-fidelity interface prototypes than can be easily redesigned. This method is aimed at rapidly uncovering bugs in the design at low cost. At the early stages of design, user interfaces often have so many bugs that this approach is productive. Cognitive engineering models form a complementary approach founded on the twin notions that prediction is a sign of understanding and control over the phenomena of interest, and a designer with an engineering model in hand can explore and explain the quantitative and qualitative effects of different design decisions before the heavy investment of resources for implementation and testing. This exploration of design space is more efficient because the choices among different design alternatives are better informed:

E1

Rather than operating solely by intuition, the designer is in a position to know which avenues are better to explore and which are better to ignore. Nearly 40 years ago, when the first textbook on cognitive psychology was written (Neisser 1967), it would have been impossible to answer the questions listed above based on psychological theory alone. Only the first of these could have been answered in a restricted way nearly 25 years ago, when the first classic monograph on the psychology of HCI was written (Card *et al.* 1983). The second, third, and fourth questions could be answered at the time of publication of first edition of the *Handbook of Applied Cognition* (Pirolli 1999). Recent progress allows us to begin to address the last three questions. The continual accumulation of knowledge and progress in predictive power is a measure of the fruitfulness of the marriage of psychology and human–information interaction.

HUMAN–INFORMATION INTERACTION

During the 1990s, there was an explosion in the amount of information that became available to the average computer user, and the development of new technologies for accessing and interacting with all of that information. The late 1980s witnessed several strands of HCI research devoted to ameliorating problems of exploring and finding electronically stored information. It had become apparent that users could no longer remember the names of all their electronic files, and it was even more difficult for them to guess the names of files stored by others (Furnas *et al.* 1987). In the mid- to late 1980s HCI literature there were several proposals to enhance users' ability to search and explore external memory. Jones (1986) proposed the Memory Extender (ME), which used a model of human associative memory (Anderson 1983) to automatically retrieve files represented by sets of keywords that were similar to the sets of keywords representing the users' working context. Latent Semantic Analysis (LSA; Dumais *et al.* 1988) was developed to mimic human ability to detect deeper semantic associations among words, like "dog" and "cat," to similarly enhance information retrieval.

Hypermedia also became a hot topic during the late 1980s, with Apple's introduction of HyperCard in 1987, the first ACM Conference on Hypertext in 1987, and a paper session at the CHI '88 conference.[2] The very idea of hypertext can be traced back to Vannevar Bush's *Atlantic Monthly* article "As We May Think" (Bush 1945). Worried about scholars becoming overwhelmed by the amount of information being published, Bush proposed a mechanized private file system, the Memex, which would augment the memory of the individual user. It was explicitly intended to mimic human associative memory. Bush's article influenced the development of Douglas Engelbart's NLS (oNLine System), which was introduced to the world in a tour-de-force demonstration at the 1968 Fall Joint Computer Conference. The demonstration of NLS – a system explicitly designed to "augment human intellect" (Engelbart 1962) – also introduced the world to the power of networking, the mouse, and point-and-click interaction. Hypertext and hypermedia research arose during the late 1980s because PC power, networking, and user interfaces had evolved to the point where the visions of Bush and Engelbart could finally be realized for the average computer user.

The confluence of increased computing power, storage, and networking and information access and hypermedia research in the late 1980s set the stage for the widespread deployment of hypermedia in the form of the World Wide Web. In 1989, Tim Berners-Lee proposed a solution (Berners-Lee 1989) to the problems that were being faced by the CERN[3] community in dealing with distributed collections of documents, which were stored on many types of platforms, in many types of formats. This proposal led directly to the release of the World Wide Web in 1990. Berners-Lee's vision was not only to provide users with more effective access to information, but also to initiate an evolving web of information that reflected and enhanced the community and its activities. The emergence of the Web in the 1990s provided new challenges and opportunities for HCI. The increased wealth of accessible content, and the use of the Web as a place to do business, exacerbated the need to improve the user experience on the Web. The phenomenal growth of the Web, and the increasing pervasiveness of interaction with electronic content provided novel questions for cognitive architectures and engineering models, and initiated the development of information foraging theory, all of which are discussed in the next sections.

COGNITIVE ARCHITECTURES

Scientific understanding and prediction in the field of HII requires integrative psychological theories. Theories need to provide predictions at multiple time-scales of phenomena and provide explanations in multiple ways. Theories also have to integrate across the typical subdivisions of psychological theory. Cognitive architectures provide such integration and are consequently a good source of applied theory for HII.

The early growth of cognitive psychology during the 1950s and 1960s was characterized by research in largely independent experimental paradigms. Each such paradigm might involve variations on one or a few experimental tasks which were designed to address a few interesting questions about the nature of cognition. In a classic challenge to the field, Allen Newell (1973) argued that cognitive psychology could not make significant progress by this divide-and-conquer approach to research. Newell believed that understanding cognition, in even simple tasks, required an integrated theory of many cognitive processes and structures. Findings from one paradigm were surely relevant to the analysis of other paradigms. Progress, Newell (1973) argued, would require the development of theories that provide a unified way of accounting for all the diverse phenomena and tasks found in the individual paradigms.

One of the significant recent developments in theories of cognitive architecture has been the integration of perceptual-motor theories with cognitive theory. Interestingly, the inspiration for this effort came from an HCI cognitive engineering model developed by Card *et al.* (1983). This section will begin with a review of the Model Human Processor developed by Card *et al.* (1983) because of the importance of this model in both HCI and as a harbinger of recent developments in cognitive architectures

The Model Human Processor

The Model Human Processor (Figure 17.1) developed by Card *et al.* (1983) is a synthesis of findings from a diverse set of cognitive psychology paradigms. The purpose for its development was to provide a way for engineers to make zero-parameter predictions about performance with HCI systems.[4] The Model Human Processor is specified as (a) a set of

E1

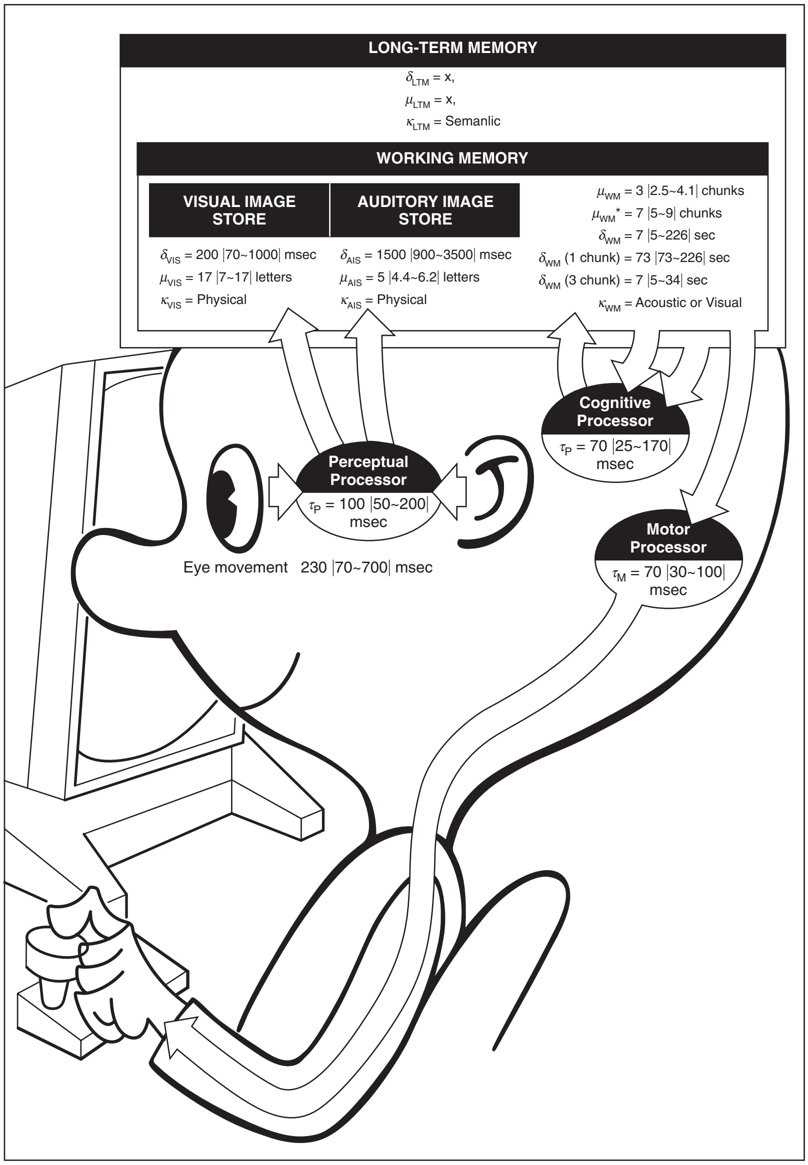


Figure 17.1 The Model Human Processor. Note that some estimates have changed since the
11 original formulation. See text for details. From S. K. Card, T. P. Moran & A. Newell (1990), *The Psychology of Human–Computer Interaction* (p. 26). Hillsdale, NJ: Lawrence Erlbaum Associates. © 1983 by Lawrence Erlbaum Associates. Reprinted with permission

memories and processors and (b) a set of principles of operation. The processors and memories are summarized schematically in Figure 17.1. There are three subsystems for perceptual input, motor action, and cognition. The processors and memories in Figure 17.1 are characterized by a set of parameters: E1

- the storage capacity in terms of maximum number of items stored;
- the decay rate of an item (the time an item will reside in memory);
- the main type of code (representation type) of information, which may be physical, visual, acoustic, or semantic; and
- the cycle time (the time at which inputs to memory are updated).

The parameters are typical values extracted from the psychological literature. The general idea is that information is input from the world through the *perceptual processors*, into the visual and auditory stores. Some of this information makes its way into a *working memory*, which is operated on by the *cognitive processor*. The cognitive processor uses associations between information in working memory and *long-term memory* to make decisions and formulate actions. Actions in working memory trigger the *motor processor* to effect behavior in the world.

Card *et al.* (1983) define a set of principles of operation for the Model Human Processor (see also Pirolli 1999). The cognitive processor works through a *recognize–act cycle* on the contents of working memory. Working memory is assumed to be the information that a person is currently heeding – their focus of attention. These working memory structures
1 2 are called *chunks* (Miller 1956; Simon 1974). In the recognize phase, information in working memory retrieves associated actions in long-term memory. The act phase of the cycle executes those retrieved actions and changes the contents of working memory. These associations between working memory information and effective actions in long-term memory are built from prior experience. The associations may be organized in the form of plans, such as plans of organized action for operating an interface. In the original Model Human Processor of Card *et al.* (1983), each recognize–act cycle takes 70 ms on average (Figure 17.1). Research since that original formulation has revised the recognize–act cycle
3 time down to 50 ms (John & Kieras 1996a).

Working memory is assumed to be of limited capacity and rapid decay rate. For instance, without rehearsal, only about 3–7 chunks of information can be held in working
4 memory for about seven seconds (Miller 1956). On the other hand, long-term memory is of very large capacity (Landauer 1986) and a very slow decay rate (Figure 17.1). It is the repository of the collective experience and learning of a person. It contains both factual knowledge as well as knowledge of how to perform procedures. Although long-term memory has effectively infinite capacity and permanent retention, there are factors that make retrieval less than perfect. These factors have to do with the ability of cues in working memory to retrieve associated information in long-term memory.

For the perceptual processor, it is assumed that auditory and visual stimuli trigger the appearance of representations in the auditory and visual memory stores. The representations in these memories encode mostly physical (non-symbolic) characteristics such as the intensity of a sound or the curvature of a line. These memories are also of limited capacity and very rapid decay (items have a half-life of 200 ms in the visual store and about 1500 ms in the auditory store). The perceptual processor has a cycle time of about 100 ms, which varies with stimulus intensity. The motor processor is assumed to operate with an approximately 70 ms cycle time. Many interactions with computers require movements of the hand or a hand-held mouse to some target location. The time to move the hand, or a mouse pointer, to a target may be calculated by Fitts' law (MacKenzie 2003), which depends on the distance to be traveled by the movement and the size of the
E1 target.

EPIC

Like the Model Human Processor, EPIC (Executive Process-Interactive Control) architecture (Kieras & Meyer 1997; Meyer & Kieras 1997a, 1997b) was developed with consideration of perceptual-motor aspects of behavior. One significant advance in EPIC is that it incorporates more recent and detailed results concerning human performance. A second significant advance is that it is actually a computer simulation system. EPIC models are constructed by specifying procedures as *production rules*. When such models are presented with the external stimuli for a task (the computer displays, keyboards, etc.) they follow the procedures for the tasks and simulate the time course of events on both the system side and human side of the HCI system.

In large part, EPIC has been developed to yield better models of attention and performance in multiple-task situations. These situations might occur in HCI, for instance, with certain computer operator jobs. Often, computer operators must coordinate and interleave their conversational tasks with a customer with database tasks with a computer. In cognitive psychology, the supervisory processes required to control and supervise other processes have traditionally been called *executive processes* – a carryover from the terminology of computer operating systems where an executive process oversees the other programs running on a computer. In EPIC, executive processes are considered to be the same as any other well-learned cognitive skill, and like other skills they are represented by production rules.

EPIC has been designed with the realization that perceptual and motor processors are complicated in their own right, and they have important interactions and constraints with cognition and executive control processes. How well people can handle multiple-task situations will depend on the structural constraints on perceptual processors, motor processors, limitations on working memory, etc. In contrast to traditional multi-tasking models, EPIC does not assume a single-channel attentional processor which must be switched from task to task, nor does it assume some limited central resource capacity on cognition. EPIC simply assumes that executive control processes must work around the structural limitations of the perceptual-cognitive-motor system. This assumption has led to successful models of attention and performance covering a large body of multi-task laboratory experiments (Kieras & Meyer 1997; Meyer & Kieras 1997a, 1997b). One of the areas of EPIC application in HII has been in understanding and making predictions about visual search over information. An application of EPIC is discussed below.

Figure 17.2 is a schematic for the EPIC architecture. It is a cognitive processor surrounded by perceptual and motor processors. The cognitive processor is controlled by production rules, and information flows through the perceptual processors, to the cognitive processor, and to the motor processors, which have feedback. To develop a specific model of a task situation requires the specification of production rules and the setting of perceptual-motor parameters. When combined with a simulator of the external environment, EPIC will simulate the serial and parallel actions required to perform the task.

The cognitive processor interacts with a working memory. Roughly, this working memory is equivalent to the short-term memory of the Model Human Processor. This working memory can also be thought of as a database that contains information representing goals and knowledge about the current state of affairs.

E1

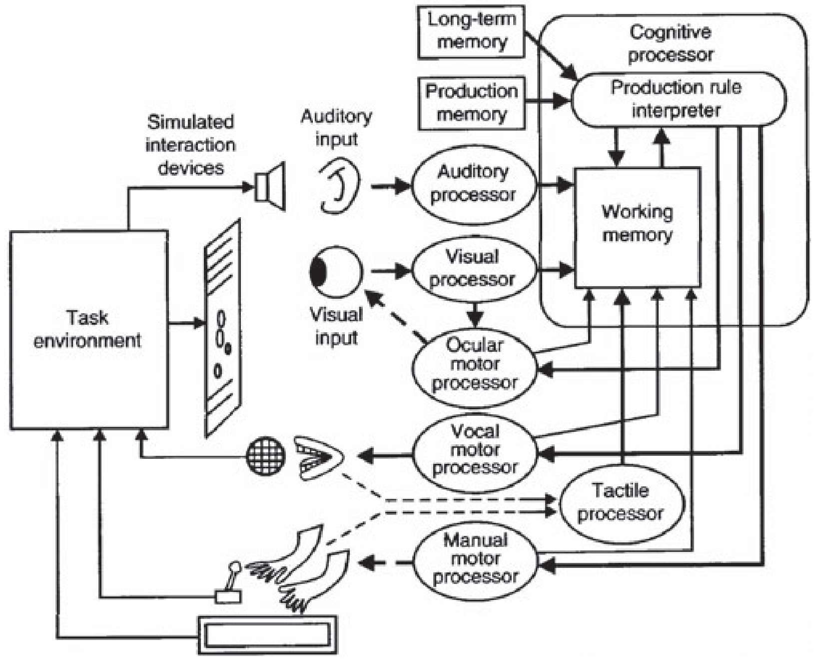


Figure 17.2 A schematic overview of the EPIC architecture. From An overview of the EPIC architecture for cognition and performance with application to human–computer interaction, *Human–Computer Interaction, 12,* 399. © 1998 by Lawrence Erlbaum Associates. Reprinted with permission

Production rules specify the flow of control of cognitive processing. Productions match to the working memory database and specify changes to the database or other actions to perform. Each production rule is of the form:

IF <condition> THEN <actions>

The condition of a rule specifies a pattern. When the contents of working memory match the pattern, the rule may be selected for application. The actions of the rule specify additions and deletions of content in working memory, as well as motor commands. These actions are executed if the rule is selected to apply.

The cognitive processor works on a 50 ms cycle. At the beginning of the cycle, working memory is updated by inputs from the perceptual processors, and by modifications that result from the application of production rules on the previous cycle. Production rules that match the contents from working memory are applied, and at the end of the cycle any motor commands that were issued are sent to the motor processor. All the production rules that match working memory are applied. The simultaneous application of production rules is a form of parallelism. Many other production systems (such as ACT-R discussed below) have limits of one production application per cycle. The perceptual and motor processors of EPIC are not necessarily synchronized with the 50 ms cognitive processor cycle. This means that perceptual inputs may have to "wait" on cognitive processing should they arrive before the beginning of the next cognitive cycle.

Working memory is partitioned into stores of different kinds of content. There are partitions of working memory for visual, auditory, and tactile information. There are slaved to the perceptual processors. That is, the outputs of those processors appear as representations in working memory. In addition, there are amodal partitions of working memory. One amodal partition is a control store that contains information about task goals and procedural steps. Another amodal partition is a general working memory that contains miscellaneous information that arises during the execution of a task.

As shown in Figure 17.2, EPIC has several perceptual processors. The time parameters associated the processing of the perceptual processors may be (a) *standard*, which come from surveys of the psychological literature and are felt to be relatively fixed across tasks, or (b) *typical*, which may vary from task to task.

There is a visual processor, which takes in stimuli from the visual scene and produces outputs in visual working memory. The visual processor has separate processors for the fovea, parafovea, and periphery. Each of these areas will produce different information from the same scene. For instance, the periphery may detect the sudden appearance of objects in the environment, the parafovea may detect the area occupied by a blob of text on a screen, and the fovea may detect the actual characters of the text. Each of these areas may have different time parameters on their processes. Event detection takes about 50 ms (periphery), while shape detection occurs about 100 ms later (parafovea), and pattern recognition about 250 ms later (fovea).

The auditory processor takes in sound stimuli and produces representations in auditory working memory. Again, different kinds of information-processing will take different amounts of time to output. The time to process a tone onset is about 50 ms, and a fully discriminated frequency appears about 250 ms later. After these outputs reach auditory working memory, they decay after about 4 s.

There are motor processors controlling the hands, eyes, and vocal tract (Figure 17.2). These can operate simultaneously. The cognitive processor sends commands to a motor processor by specifying the type and parameters of the movement to perform. The motor processor then translates these into a simulated movement. Movements are specified in terms of features. The time to execute a movement depends on movement features and the mechanical properties of the movements (e.g., the trajectory of movement of the hands).

The motor processors work in two phases: *preparation* and *execution*. In the preparation phase, a command is received from the cognitive processor and recoded into a set of movement features. For instance, to specify the tap of a finger on a key may require five features: the tap style, hand, finger, direction of movement, and extent of movement. The generation of each feature takes 50 ms, but features may be reused from previous movements, or generated in advance. For instance, tapping two different keys will share some features and allow the re-use of features. Tapping the same key twice will re-use all the features. If a movement can be anticipated, then the features can be prepared in advance. The execution phase has a delay of 50 ms to initiate the movement specified in the preparation phase. The physical movement depends on mechanical properties. For instance, the tap motion of a finger on the keyboard depends on a version of Fitts' law. The manual processor is capable of different movement styles such as punching keys, tapping, two-fingered patterns, pointing with a mouse, or pointing with a joystick. The occulomotor system has both voluntary motions (saccades) and involuntary (reflexive) motions. The vocal processor is capable of simple fixed utterances.

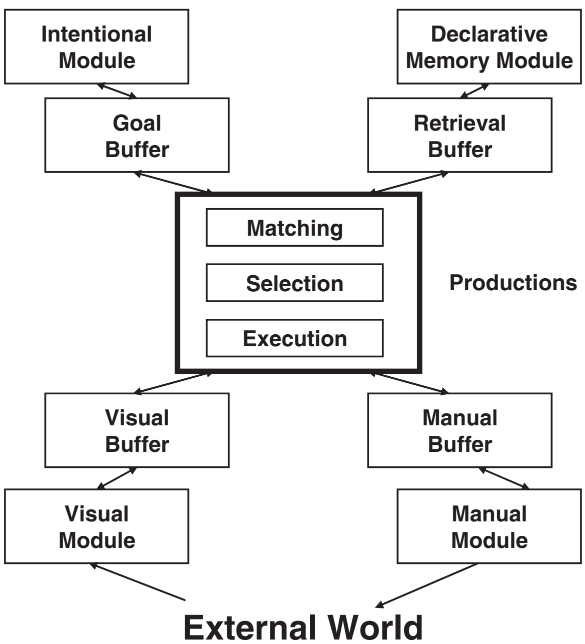
E1

ACT

The ACT family of production system theories has the longest history of production system cognitive architectures. The seminal version of the ACT theory was presented in Anderson (1976), and it has undergone several major revisions since then (Anderson 1983, 1990, 1993; Anderson & Lebiere 1998; Anderson *et al.* 2004). Until recently, it has been primarily a theory of higher cognition and learning, without the kind of emphasis on perceptual-motor processing found in EPIC (Kieras & Meyer 1997) or the Model Human Processor (Card *et al.* 1983). The success of ACT as a cognitive theory has been historically in the study of memory (Anderson & Pirolli 1984; Anderson & Milson 1989), language (Anderson 1976) problem-solving (Anderson 1993) and categorization (Anderson 1991). As a learning theory, ACT has been successful (Anderson 1993) in modeling the acquisition of complex cognitive skills for tasks such as computer programming, geometry, and algebra, and in understanding transfer of learning across tasks (Singley & Anderson 1989). ACT has been strongly tested by application in the development of computer tutors (Anderson *et al.* 1990).

ACT-R, like previous versions of the ACT theory, contains assumptions about (1) knowledge representation, (2) knowledge deployment (performance), and (3) knowledge acquisition (learning). The current publicly released version of the architecture is ACT-R 5.0, which is illustrated in Figure 17.3. The architecture is arranged as a set of modules, each devoted to processing a particular kind of information, which are integrated and coordinated through a centralized production system module. Each module is assumed to

6



E1

Figure 17.3 The ACT-R 5.0 architecture deposit information into buffers associated with the module, and the central production system can only respond to the contents of the buffers, not the internal workings of the modules. This is consistent, for instance, with the observation that people do not have awareness of all of the retinal information in the visual field. The ACT-R 5.0 theory makes no claim about the complete set of modules that may eventually be identified. Each module, including the production module, has been hypothesized to occur in particular brain locations:

- Visual module (occipital cortex, and others) and visual buffers (parietal cortex) are based on EPIC's modules and keeps track of objects and locations in the visual field.
- Manual module (motor cortex; cerebellum) and manual buffer (motor cortex) are based on EPIC's modules and is associated with control of the hands.
- Declarative module (temporal lobe; hippocampus) and retrieval buffer (ventrolateral prefrontal cortex) are associated with the retrieval and awareness of information from long-term declarative memory.
- Goal buffer (dorsolateral prefrontal cortex) keeps track of the goals and internal state of the system in problem-solving.
- Production system (basal ganglia) is associated with matching the contents of module buffers and coordinating their activity. The production includes components for pattern matching (striatum), conflict resolution (pallidum), and execution (thalamus). A production rule can be thought of as a formal specification of the flow of information from buffered information in the cortex to the basal ganglia and back again.

Historically, ACT-R provided limited, ad hoc modeling of perceptual-motor behavior. The production, declarative, and goal modules in ACT-R 5.0 are vestigial remnants of those earlier versions of ACT-R, and still remain the core of the current architecture. The declarative memory module and production system module store and retrieve information
5 that corresponds to *declarative knowledge* and *procedural knowledge* (Ryle 1949). Declarative knowledge is the kind of knowledge that a person can attend to, reflect upon, and usually articulate in some way (e.g., by declaring it verbally or by gesture). Declarative knowledge includes the kinds of factual knowledge that users' can verbalize, like "The 'open' item on the 'file' menu will open a file." Procedural knowledge is the know-how we display in our behavior, without conscious awareness. For instance, knowledge of how to ride a bike and how to point a mouse to a menu item are examples of procedural knowledge. Procedural knowledge specifies how declarative knowledge is transformed into active behavior. Declarative knowledge in ACT-R is represented formally in terms of
6 7 chunks (Miller 1956; Simon 1974). Whereas the information in the declarative memory module corresponds to personal episodic and semantic knowledge that promotes long-term coherence in behavior, the goal module stores and retrieves information that represents the internal intention and problem-solving state of the system and provides local coherence to behavior.

Chunks are retrieved from long-term declarative memory by an activation process. Activation may be interpreted metaphorically as a kind of mental energy that drives cognitive processing. Activation spreads from the current focus of attention, including goals, through *associations* among chunks in declarative memory. These associations are built up from experience, and they reflect how ideas co-occur in cognitive processing. Generally, activation-based theories of memory predict that more activated knowledge structures will receive more favorable processing. The spread of activation from one cognitive E1 structure to another is determined by weighting values on the associations among chunks. These weights determine the rate of activation flow among chunks.

While sharing many commonalities with the EPIC perceptual-motor modules, ACT-R 5.0 does differ in several respects. The visual system in ACT-R 5.0 is separated into two components: (1) a *where* system that processes locations in the visual field, and a (2) *what* system that processes objects. Productions may request information from the "where" system by specifying a set of constraints based on visual properties (e.g., "an object that is colored red") or spatial location (e.g., "an object at the top of the screen") and the "where" system will return a chunk that matches those constraints. This supports the modeling of pre-attentive pop-out effects (Treisman & Gelade 1980) that occur, for example, when a display includes one red object amongst a set of green objects. In such a case, a production rule that requests "objects that are colored red" will cause the "where" system to return a chunk specifying the single red object displayed, and visual search time will be constant regardless of the number of green objects on the display. On the other hand, a production rule request for "an object that is colored green" will cause the "where" system to return a chunk that represents any one of the green objects, and the time to search for a particular green object will require repeated calls to the "where" system (a serial self-terminating search) that will exhibit time costs that depend on the number of other green objects in the display.

The "what" system of the visual module keeps track of visual objects. Production rules may request the "what" system to identify objects at a location, which causes the "what" system to shift visual attention to that location and return a declarative chunk that represent the object. ACT-R 5.0 supports a coarse model of visual attention in which visual search occurs at a rate of 185 ms per visual item, and a fine-grained model called EMMA (Salvucci 2001) in which the time for eye movements have time costs related to the eccentricity between the current focus of attention and the target location requested by a production.

Production rules are used to represent procedural knowledge in ACT-R. That is, they specify how to apply cognitive skill (know-how) in the current context, and how to retrieve and modify information in the buffers to other modules. Like EPIC, production rules in ACT-R have the basic IF<condition> THEN <actions> format. In ACT-R, each production rule has conditions that specify structures that are matched in buffers corresponding to information from the external world or other internal modules. Each production rule has actions that specify changes to be made to the buffers. As in EPIC, it is assumed that the cycle of production matching and execution takes about 50 ms to complete.

ACT-R 5.0 is a mix of parallel and serial processing. Modules may process information in parallel with one another. So, for instance, the visual modules and the motor modules may both operate at the same time. However, there are two serial bottlenecks in process. First, only one production may execute during a cycle (which is different than EPIC). Second, each module is limited to placing a single chunk in a buffer.

ACT-Scent

Seeking and gathering information for some purpose typically requires that users perform some mix of navigation through on-line information structures and the use of search engines. To model these activities requires a theory of how people perceive those informa-

tion structures and then decide on the best course of action to take. An elaboration of the ACT-R architecture has been developed to model such activities. This architecture, called
8 ACT-Scent[5] (Pirolli in press), was developed within a more general theoretical framework called Information Foraging Theory. This theory was developed to understand and predict information seeking and gathering behavior with complex technologies. Among other applications, Information Foraging Theory (Pirolli & Card 1999) has been used to develop cognitive models of Web navigation (SNIF-ACT 1.0 and 2.0 described below) that form the basis for a system that predicts Web usability. The theory is concerned with human behavior and technology involved in gathering information for some purpose, such as making a medical decision, finding a restaurant, or solving a programming problem.

Information Foraging Theory

Information Foraging Theory has adopted the *rational analysis* program initiated by Anderson (1989, 1990, 1991). The rational analysis approach involves a kind of reverse engineering in which the theorist asks (a) *what* environmental problem is solved, (b) *why* is a given behavioral strategy a good solution to the problem, and (c) *how* is that solution realized by cognitive mechanism. The products of this approach include (a) characterizations of the relevant goals and environment, (b) mathematical rational choice models (e. g., optimization models) of idealized behavioral strategies for achieving those goals in that environment, and (c) computational cognitive models. This methodology is founded on the heuristic assumption that evolving, behaving systems are well-designed (rational) for fulfilling certain functions in certain environments. Rational analysis is a variant form of an approach called *methodological adaptationism* that has also shaped research programs in behavioral ecology (e.g., Tinbergen 1963; Mayr 1983; Stephens & Krebs 1986), anthropology (e.g., Winterhalder & Smith 1992), and neuroscience (e.g., Glimcher 2003).

Rational Analysis

Anderson has used rational analysis to study the human cognitive architecture by assuming that natural information-processing mechanisms involved in such functions as memory (Anderson & Milson 1989; Anderson & Schooler 1991) and categorization (Anderson 1991) were well-designed by evolutionary forces to meet the problems posed by the environment. The key assumption behind rational analysis could be stated as

> *Principle of rationality*: The cognitive system optimizes the adaptation of the behavior of the organism.

As developed by Anderson (1990) rational analysis requires a focus on understanding the structure and dynamics of the environment. This understanding provides a rationale for the design of information-processing mechanisms. Anderson proposed the following recipe for rational analysis:

1. Precisely specify the goals of the agent.
2. Develop a formal model of he environment to which the agents is adapted. E1

3. Make minimal assumptions about the computational costs.
4. Derive the optimal behavior of the agent considering (1)–(3).
5. Test the optimality predictions against data.
6. Iterate.

Note, generally, the emphasized focus on optimal behavior under given goals and environmental constraints and the minimal assumptions about the computational structure that might produce such behavior.

Interaction with the information environment differs in a fundamental way from well-defined task environments that have been the dominant paradigms in HCI, such as expert text editing (Card *et al.* 1983) or telephone assistance (Gray *et al.* 1993). In contrast to such tasks in all but the most trivial cases, the information forager must deal with a *probabilistically textured* information environment (Brunswik 1952). In contrast to application programs such as text editors and spreadsheets, in which actions have fairly determinate outcomes,[6] foraging through a large volume of information involves uncertainties, for a variety of reasons, about the location, quality, relevance, veracity, and so on, of the information sought and the effects of foraging actions. The ecological rationality of information foraging behavior must be analyzed through the theoretical lens and tools appropriate to *decision making under uncertainty*. The determinate formalisms and determinate cognitive mechanisms that are characteristic of the HCI paradigm are inadequate for the job of theorizing about information foraging in probabilistically textured environments. Models developed in Information Foraging Theory draw upon probabilistic models, and especially Bayesian approaches, and they bear similarity to economic models of decision-making (rational choice) under uncertainty and engineering models.

ACT-Scent Architecture

Figure 17.4 presents the basic ACT-Scent architecture used in information foraging models (Pirolli 1997, 2005; Pirolli & Card 1999). It couples a simpler version of the ACT-R architecture to a module that computes *information scent*. Below, this chapter will present specific models of Web foraging (SNIF-ACT 1.0 and SNIF-ACT 2.0) developed within this architecture. The architecture includes a declarative memory containing chunks, a procedural memory containing production rules, and a goal memory containing the hierarchy of intentions driving behavior. The information scent module is a new addition to ACT that is used to compute the utility of actions based on an analysis of the relationship of content cues from the user interface to the user's goals.

A Spreading Activation Model of Information Scent

Information foraging behavior will often depend on assessments of the utility and costs of pursuing information items. In browsing for information on the Web, people must base navigation decisions on assessments of information scent cues associated with links from one Webpage to another. These information scent cues are the small snippets of text and graphics that are associated with Web links. Those cues are intended to represent tersely the content that will be encountered by choosing a particular link on one page and

E1

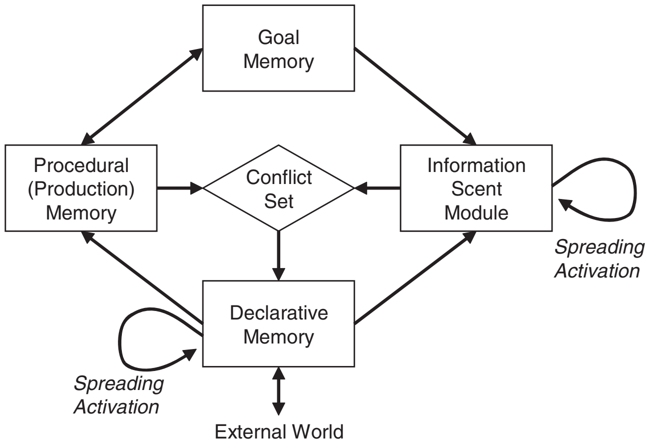


Figure 17.4 The ACT-Scent cognitive architecture

navigating to the linked page. When browsing the Web by following links, users must use these cues presented proximally on the web-pages they are currently viewing in order to make navigation decisions. The measure of information scent provides a means to predict how users will evaluate different links on a web-page, and as a consequence, the likelihood that a particular link will be followed.

The rational analysis of the use of information scent assumes that the goal of the information forager is to use proximal external information scent cues (e.g., a web-link) to predict the utility of distal sources of content (i.e., the web-page associated with a web-link), and to choose to navigate the links having the maximum expected utility. Pirolli (2005) decomposed this problem into three parts: (1) a Bayesian analysis of the expected relevance of a distal source of content conditional on the available information scent cues; (2) a mapping of this Bayesian model of information scent onto a mathematical formulation of spreading activation; and (3) a model of rational choice that uses spreading activation (Anderson & Pirolli 1984) to evaluate the utility of alternative choices of web-links. This rational analysis yielded a spreading activation theory of utility and choice.

The spreading activation theory of information scent assumes that the user's cognitive system represents information scent cues and information goals in cognitive structures called *chunks*. Figure 17.5 presents a schematic example of the information scent assessment subtask facing a Web user. Figure 17.5 assumes that a user has the goal of finding information about "medical treatments for cancer," and encounters a web-link labeled with the text that includes "cell," "patient," "dose," and "beam." The user's cognitive task is to predict the likelihood that a distal source of content contains desired information based on the proximal information scent cues available in the Web link labels. Each node in Figure 17.5 represents a cognitive chunk. Chunks representing information scent cues are presented on the right side of Figure 17.5, chunks representing the user's information need are presented on the left side. Also represented by lines in Figure 17.5 are *associations* among the chunks. The associations among chunks come from past experience. The strength of associations reflects the degree to which proximal information scent cues

E1

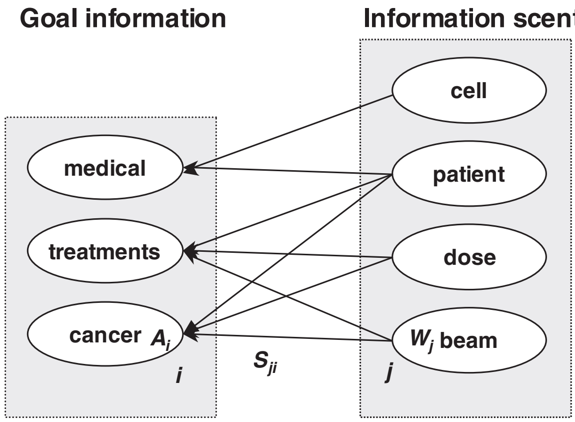


Figure 17.5 A cognitive structure in which cognitive chunks representing an information goal are associated with chunks representing information scent cues from a web-link

predict the occurrence of unobserved features. The strength of association between a chunk *i* and chunk *j* is computed as,

$$S_{ji} = \log\left(\frac{\Pr(i|j)}{\Pr(i)}\right) \tag{17.1}$$

Where Pr(*i*|*j*) is the probability (based on past experience) that chunk *i* has occurred when chunk *j* has occurred in the environment, and Pr(*i*) is the base rate probability of chunk *i* occurring in the environment. Equation 17.1 is also known as *Pointwise Mutual Information* (Manning & Schuetze 1999) or PMI.

It is assumed that when a user focuses attention on a web-link their attention to information scent cues activates corresponding cognitive chunks. Activation spreads from those attended chunks along associations to related chunks. For instance, activation would flow from the chunks on the right of Figure 17.5 through associations to chunks on the left of Figure 17.5. The amount of activation accumulating on the representation of a user's information goal provides an indicator of the likelihood that a distal source of information has desirable features based on the information scent cues immediately available to the user. For each chunk *i* involved in the user's goal, the accumulated activation received from all associated information scents chunks *j* is,

$$A_i = \sum_j W_j S_{ji} \tag{17.2}$$

where W_j represents the amount of attention devoted to chunk *j*. The total amount of activation received by all goal chunks *i* is just,

$$V = \sum_i A_i \tag{17.3}$$

The theory assumes that the utility of choosing a particular link is just the sum of activation it receives (Eqn 17.3) plus some random noise. From this assumption (see, Pirolli

2005) one can derive that the probability that a user will choose link L, having a summed activation V_L, from a set of links C on a Webpage, given an information goal, G, is

$$\Pr(L|G,C) = \frac{e^{\mu V_L}}{\sum_{k \in C} e^{\mu V_k}} \tag{17.4}$$

The information scent mechanisms are integral to modeling user judgments of which navigation action to take, including when to give up.

APPLICATIONS

This section illustrates the application of computational cognitive models to HII. This set of examples was selected to illustrate each of the three theories discussed above in three HII application domains. The examples include an EPIC model of visual search over information displays, ACT-R models of hand-held devices, and ACT-Scent models of seeking information on the Web.

EPIC Application: Visual Search of Hierarchical Information Displays

Many tasks involving interaction with information require visual navigation through displays of data that are hierarchically organized. Simple examples of such displays include menus (Hornof & Kieras 1997; Byrne *et al.* 1999) and web-pages. For instance, web-pages often arrange sets of links into groups (i.e., links within a group are proximal to one another and groups are spaced apart from one another), and provide distinctive labels for the visual groups. For instance, the home page for an on-line newspaper may contain visually grouped links labeled "Headlines," "Local," "Business," "Sports," "Entertainment," and so on. The benefits of such hierarchical arrangements are common wisdom in the design world (Spool *et al.* 1999).

Hornof and Halverson (2003) developed an EPIC model to make quantitative predictions regarding visual navigation of such hierarchically arranged information displays and to make predictions about the visual scan paths of users. Hornof and Halverson presented labeled and unlabeled layouts of one, two, four, or six groups of five text labels (pseudo-words). Figure 17.6 shows examples of a six-group labeled layout used in the experiment. Participants were visually presented a pseudo-word that would be their target for visual search, then they were presented with a display to search. In conditions in which labeled groups were presented, the users were presented with the group label from the group in which the target would be found, as well as the pseudo-word to find within the group. The location of the target in the display was random across trials.

Hornof and Halverson (2003) encoded a representation of the task and display into EPIC, along with the visual-perceptual features of the screen objects. Production rules were written to represent the cognitive strategies of study participants. Two strategies were found to fit the data:

- *Noisy-systematic search strategy* was used to predict data for the unlabeled layouts. This strategy assumes that people make a maximally efficient foveal sweep of the

E1

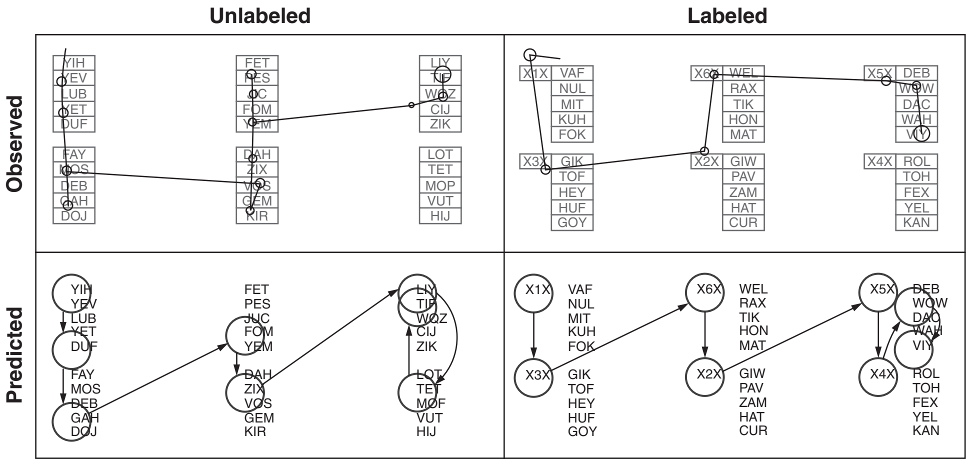


Figure 17.6 Observed fixation patterns (top) vs. those predicted by EPIC (bottom) for a six-group unlabeled display trial (left; modeled using a noisy-systematic strategy) and a six-group labeled display trial (right; modeled using a mostly systematic two-tiered search strategy). Circle diameters represent fixation duration in the observed data and represent foveal region in the predicted data

display in which minimizes the number of fixations required to fixate the contents of the display in the high-resolution fovea (Hornof & Kieras 1997). The strategy is noisy because it sometimes overestimates how far to move the eyes.

- *Mostly systematic two-tiered search strategy* was used to predict eye movements on displays containing labeled groups. This strategy assumes that people first search labels until a target group is found, then search within that group. It is mostly systematic because it searches in an ordered fashion 75 per cent of the time, and random order 25 per cent of the time.

Figure 17.6 shows observed eye fixations patterns above eye movement patterns predicted by EPIC. One can observe good qualitative match of the model's search strategy to those of a participant. Figure 17.7 shows the fit of the model's prediction to observed visual search times. The model predicts the unlabeled visual search with an average absolute error of 8 per cent and the labeled visual search with an average absolute error of 6 per cent.

ACT-R Application: Information Seeking on a PDA

A cognitive engineering tool called CogTool (John *et al.* 2004; Luo & John 2005) has been developed to support the rapid development of a class of simple ACT-R models, called ACT-Simple, that correspond to the KLM (keystroke-level model) of Card *et al.* (Card *et al.* 1983). The scope of KLMs is limited to capturing the error-free performance of highly skilled users executing a single specific method on a given interface. Task execution is modeled in terms of time parameters associated with (a) the physical operators for

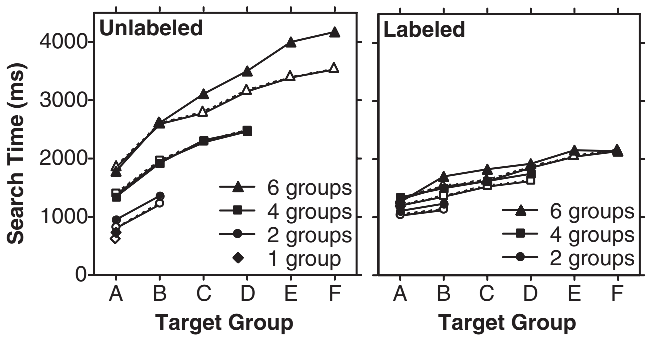


Figure 17.7 Observed search times (solid lines) and predicted (dashed lines) by the EPIC models using the noisy-systematic strategy (for unlabelled displays; left) and mostly systematic two-tiered search strategy (for labeled displays; right)

keystrokes, homing (moving the hands to a home location), drawing, (b) a generic mental operation (to capture time needed to think), and (c) system response time.

The development of CogTools has been guided by a set of principles aimed at accelerating the spread of cognitive engineering among user interface designers (John *et al.* 2004): (1) exploit tools already in widespread use among the design and cognitive modeling communities; (2) tightly couple design mockup tools to cognitive modeling tools so that changes to an user interface design are immediately reflected in changes to predictions of the cognitive model; and (3) avoid the need for programming by using such techniques as programming by demonstration.

CogTool allows a user interface designer to model an existing UI mocked up as existing web-pages. Alternatively, the designer may mock up a UI as web-pages using a web design application plus a palette of standard interface widgets such as buttons, check-boxes, text fields, pull-down menus, cascading menus, roll-over images, simulated speech-I/O, links, etc. The user interface mock-up defines the UI layout, as well as the effects of user actions (such as clicking on a button). The designer then may demonstrate a method of using the UI while recording the interaction with a tool called the Behavior Recorder. If there are alternative methods for performing some goal with the UI, then each method must be demonstrated. The resulting trace of behavior is assumed to represent an expert error-free execution of a given method for achieving some goal with the UI.

Each demonstration of a method with the UI is sent to a compiler that maps the behavior into a set of production rules that would generate the same behavior. These production rules implement a *press-key* ACT-Simple command that takes ~200–350 ms, a *move-mouse* ACT-Simple command that takes ~650–750 ms to move 500 pixels, a *move-hand* command that takes ~650–800 ms to execute, a *look-at* command that takes 150 ms, and a *think* command that takes 1200 ms. Most ACT-Simple commands compile to one production, and the *look-at* command compiles to two productions (one for shifting visual attention and one for ensuring encoding of a visual object).

Luo and John (2005) used CogTool to make predictions about use of the Palm Vx handheld PDA with an application that provides a city guide for New York City. The goal was to model performance times for alternative methods for using the city guide to find the open hours for the Metropolitan Museum of Art (MET). A Web-based storyboard mockup of the interface states and transitions that would result from a user performing these methods was created. An analyst demonstrated each method on the mockup using the Behavior Recorder. The demonstrated behavior trace was automatically compiled into ACT-Simple commands, which in turn were compiled into ACT-R production rules. The resulting ACT-R models were then run to provide the equivalent of a KLM performance prediction.

The methods analyzed were chosen to cover a range of interaction techniques possible with the Palm PDA. Four methods were studied: (1) map navigation, which involved clicking on increasingly more detailed maps to zoom in on the location of the MET; (2) soft keyboard, which involved tapping on a layout of alphanumeric characters to type in queries; (3) graffiti, which involved using a stylus to enter characters using the graffiti shorthand technique; and (4) scroll bar, which involved scrolling through a list of museums until the MET appeared. Users (*N* = 10) were asked to perform each of he methods ten times on Palm PDAs, following a practice phase. The average task times for the methods ranged from 9.00 s to 13.60 s, with map navigation being the fastest method and Graffiti being the slowest. The average CogTool prediction error for the four methods was 3.7 per cent (ranging from 1.38 to 7.43 per cent).

ACT-Scent Application: Seeking Information on the Web

The rational analysis of information scent presented above can be used to develop models of how users choose links on the Web. A similar rational analysis developed in Pirolli (2005) can be used to predict when users will leave a website. SNIF-ACT is a computation model developed in the ACT-Scent architecture based on these rational analyses. This section presents an overview of the SNIF-ACT model (Pirolli & Fu 2003; Pirolli 2005), as well as an automated Web usability system called Bloodhound (Chi *et al.* 2003) that was inspired by SNIF-ACT. For comparison, this chapter discusses a very similar Web usability analysis method called Cognitive Walkthrough for the Web (Blackmon *et al.* 2005; Kitajima *et al.* 2005) that is also based on the concept of information scent.

SNIF-ACT

SNIF-ACT assumes that users have the procedural knowledge necessary to use the browser, such as clicking on a link, or clicking on the “back” button to go back to the previous web-page. This procedural knowledge is represented as a set of production rules. SNIF-ACT also assumes that users have knowledge of the addresses of most popular Web search engines. This knowledge is represented as chunks in declarative memory.

In a SNIF-ACT simulation, information scent cues on a computer display activate chunks and activation spreads through the declarative network of chunks. The amount of activation accumulating on the chunks matched by a production is used to compute a utility score, which is used to evaluate and select productions. For instance, the utility of productions implementing the selection and clicking of links is based on the activation that spreads from the link that the productions match against.

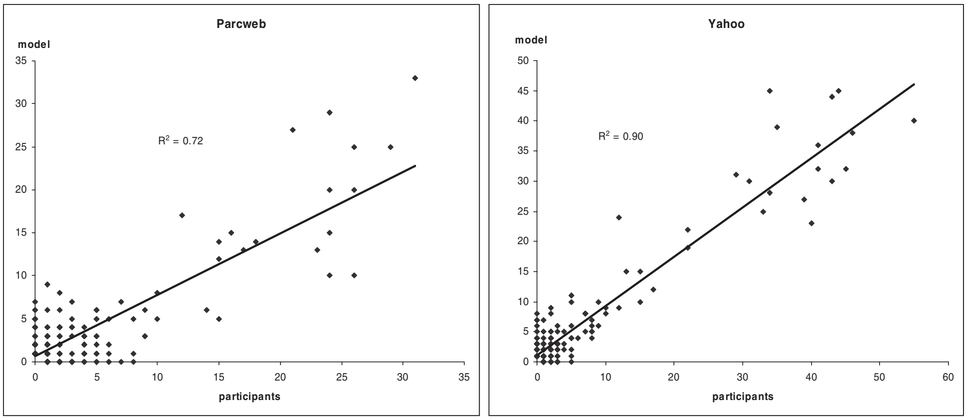


Figure 17.8 The scatter plots for the number of times links were selected by the model and by user at the Parcweb and Yahoo sites (eight tasks per site)

SNIF-ACT 1.0 (Pirolli & Fu 2003) was used to simulate four users working on two tasks each. Users were free to navigate anywhere on the Web to accomplish these tasks (for details, see Card *et al.* 2001). SNIF-ACT 2.0 (Pirolli 2005) was matched to data from 244 users. Users could work on eight tasks on two websites. Users were constrained to never leave the given website when performing their given task. Monte Carlo simulations with the model were used to generate data for the 16 tasks. For each task, the number of times SNIF-ACT was run was equivalent to the number of users observed on the task. Each point in Figure 17.8 plots data for a single link on a website, and each data point represents the number of users who selected that link, and the number of times SNIF-ACT selected the same link. Figure 17.8 shows that SNIF-ACT 2.0 provides good match to the data.

SNIF-ACT is a computational model derived from the rational analyses of Web navigation. The major assumption of the model is that Web navigations can be characterized by mechanisms that maximize expected information gain. Expected information gain is estimated by a spreading activation mechanism that calculates the relatedness of information goal and link text. The good fits to human data provide strong support for the use of information scent to characterize information-seeking decisions on the Web.

Bloodhound

The Bloodhound service (Chi *et al.* 2003) employs a Web user flow model to predict website usage patterns and identify Website navigation problems. The service employs a variation on a graph flow algorithm which abstracts away from the details of the SNIF-ACT model. This assumes that users come to a website with some information goal and forage for information by choosing links based on proximal information scent cues.

Figure 17.9 presents an overview of the process used by the Bloodhound service. A person (the *website analyst*) interested in performing a usability analysis of a website must indicate the website to be analyzed, and provide a text description representing a task that

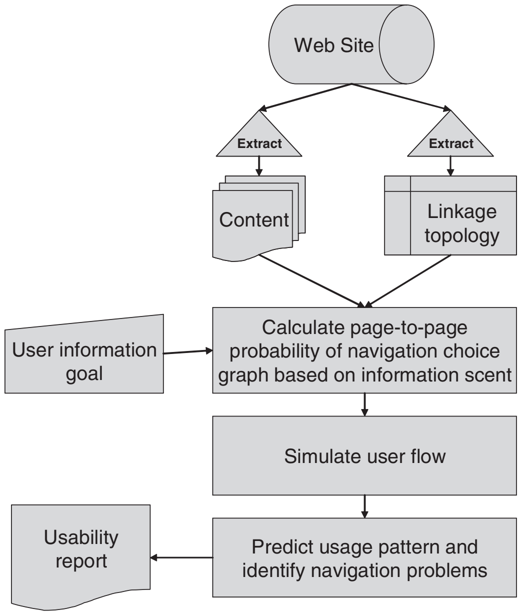


Figure 17.9 The conceptual flow chart for the processing done by the Bloodhound Web usability service

users are expected to be performing at the site. Bloodhound then must trawl the website to develop a representation of the linkage topology (the page-to-page links) and download the web-pages (content). From these data, Bloodhound analyzes the web-pages to determine the proximal information scent cues associated with every link on every page. At this point Bloodhound essentially has a representation of every page-to-page link, and the proximal cues associated with that link. From this, Bloodhound develops a graph representation in which the nodes are the website pages, the vertices are the page-to-page links at the site, and weights on the vertices represent the probability of a user choosing a particular vertex given the user's information goal and the proximal information scent cues associated with the link (e.g., Eqn 17.4). This graph is represented as a page-by-page matrix in which the rows represent individual unique pages at the site, the columns also represent website pages, and the matrix cells contain the navigation choice probabilities that predict the probability that a user with the given information goal, at a given page, will choose to go to a linked page. Using matrix computations (similar to those used in modeling a Markov process), this matrix is used to simulate user flow at the website by assuming that the user starts at some given webpage and iteratively chooses to go to new pages based on the predicted navigation choice probabilities. The user flow simulation yields predictions concerning the pattern of visits to web-pages, and the proportion of users that will arrive at target web-pages contain the information relevant to their tasks.

The Bloodhound service is provided over the Web. An input screen is provided to website analysts that allows them to enter specifications of user tasks, the website URL, and the target pages that contain the information relevant to those tasks. An analysis is then performed by Bloodhound and a report is then automatically generated that indicates such things as the predicted number of users who will be able to find target information relevant to the specified task, and intermediate navigation pages that are predicted to be highly visited that may be a cause of bottlenecks.

Chi *et al.* (2003) performed an evaluation of the capability of Bloodhound to predict actual user navigation patterns. Users were solicited to perform Web tasks at home, office, or place of their choosing and their performance was logged using a remote usability testing system. Four different types of website were studied with eight tasks of varying difficulty for each site. The comparison of interest was the match between observed and predicted usage patterns for each task and website. For each task + website, the observed data were the distribution of the frequency of page visits over every webpage. For instance, for a particular task + website, the home page might be visited 75 times, another page 25 times, and so on. The comparison was the distribution of page visits for that task and website as predicted by Bloodhound. Of the $4 \times 8 = 32$ combinations of websites and tasks, there were strong correlations (Pearson $r > 0.8$) of observed and predicted visitation frequencies for twelve cases, moderate correlations ($0.5 \leq r \leq 0.8$) for 17 cases, and weak correlations ($r < 0.5$) for three cases. Given that this was the first evaluation of Bloodhound the results seemed like a validation of the promise of the approach.

Cognitive Walkthrough for the Web

Cognitive Walkthrough for the Web (CWW) is a semi-automated method for finding and repairing Web usability problems that is similar in many ways to Bloodhound. CWW derives from a cognitive model called CoLiDes (Comprehension-based Linked model of Deliberate Search; Kitajima *et al.* 2000). Although CoLiDes differs in detail from SNIF-ACT, it shares the basic assumptions that the Web user has an information goal and information scent drives information-seeking behavior. CoLiDes uses a technique called Latent Semantic Analysis (LSA) to compute information scent. LSA (Landauer & Dumais 1997) assumes that the meaning of a word is associated with the meanings of all contexts that it has occurred in, while simultaneously the meaning of a message is associated with all the words it contains. In practice, the meaning of words is computed from a corpus of documents assumed to represent the linguistic environment of some population (e.g., college students). A word-by-document matrix, in which cell entries indicate the occurrence of some particular word in a particular document in the corpus is then extracted (typically, the log frequency of a word in a document), and submitted to a singular value decomposition (SVD) that is similar to factor analysis. This procedure computes a set of dimensions (substantially less than the number of documents) that may be interpreted to represent latent semantic factors. Each word can then be represented by its position in a space defined by these semantic factors. That is, each word will be represented as a vector of scores that indicate a position on each latent semantic factor. The psychological similarity of two words is just a distance measure between them in the semantic space defined by these latent factors. Dumais (2003) provides a survey of LSA applications in psychology and other fields. Turney (2001) has shown that PMI and LSA give very similar results on synonym tests. CoLiDes assumes that users' goals are represented as a collection of words, and links and web-page headings are represented by the text from the web-page, plus elaborations that consist of words that have high similarity to that text. E1

CWW identifies four kinds of problems, proposes repairs to those problems, and predicts the severity of those problems before and after repairs. The procedure (Blackmon *et al.* 2005) consists of:

1. Selecting a semantic space (generated from an appropriate document corpus) to represent a given user group.
2. Constructing a representative set of user goals for the website.
3. Simulate how users will parse a web-page into regions and sub-regions.
4. Simulate the elaboration of link text and heading text.
5. Apply LSA to obtain similarity scores from goals to Web headings and links. Mark the correct heading and link that should be chosen for each given goal.
6. Apply a set of problem identification rules.
7. Apply suggested repairs.
8. Apply problem severity formulae to predict severity for repaired and unrepaired problems.

The problem identification rules identify:

- *Weak link scent*, in which no correct links have strong similarity scores with the goal.
- *Unfamiliar links or headings*, which are indicated by LSA measures from the representative semantic space.
- *Competing headings*, which occur when a heading or associated sub-region on the web-page shows high similarity to the user's goal, but does not contain a correct link.
- *Competing links* in which there are links, other than the correct ones, that have high similarity to the user's goal.

Repairs to these problems typically involve substituting words to reduce the problem (e.g., by substituting words with high similarity to the goal in correct links with weak scent). Problem severity is predicted by a regression formula fit to data from a wide variety of tasks. The problem severity formula predicts the number of clicks required to get to a desired page on the analyzed web-page based on the familiarity of the correct link, the strength of scent of the correct link, and the number of competing links under competing headings.

GENERAL DISCUSSION

Computational cognitive models have been applied to an ever-broadening range of problems in human–computer interaction (Pirolli 1999). Recent applications in human–information interaction suggest that this record of success will continue. Cognitive architectures, such as EPIC, ACT-R, and ACT-Scent provide an integrated approach to modeling perception, motor action, and higher-order cognition. Their utility has been demonstrated in generating cognitive engineering models for such applications as visual search, mobile computing, and Web use.

Although working with such cognitive models directly requires some amount of training, the CogTool project (John *et al.* 2004; Luo & John 2005) suggests that standard rapid-prototyping tools for user interface design can be enhanced with built-in cognitive engineering evaluations. Similarly, the Bloodhound system (Chi *et al.* 2003) was targeted for use by Web designers with no training in cognitive engineering. The concept of

E1

information scent has been used to develop Web usability guidelines (Spool *et al.* 2004) and evaluation methods (Blackmon *et al.* 2002, 2005) for use by practitioners.

One of the significant challenges that lie ahead for cognitive models of HII is the problem of modeling the interpretation of content into actionable knowledge. For text, this involves using statistical language techniques such as PMI or LSA to build associative networks that support the mapping of external text onto user goals and procedural knowledge (e.g., the selection of production rules). For multimedia, this is much more difficult. Can we model how people make sense of the multimedia content with which they interact? This requires the integration of additional components for robust language comprehension, graphics understanding, rich knowledge representation, reasoning, knowledge acquisition, and meta-cognition, among other things. Many of these components have a long history of research in artificial intelligence and computational linguistics, but have yet to be incorporated into an integrated cognitive architecture such as the ones described in this chapter.

ACKNOWLEDGMENTS

The writing of this chapter has been supported in part by MDA 904-03-C-0404 to Stuart K. Card and Peter Pirolli.

NOTES

1 The term "human–information interaction" first appeared in the public literature in a birds-of-a-feather meeting title in 1995: Gershon, N. (1995, December). Human Information Interaction. In the *Proceedings of the Fourth International World Wide Web Conference* (http://www.w3.org/Conferences/WWW4/bofs/hii-bof.html).
2 The ACM is the Association for Computing Machinery, which is the main computer science organization, and CHI is the conference for the Computer Human Interaction special interest group of the ACM.
3 CERN is the European Organization for Nuclear Research, a multinational science community operating the world's largest particle physics laboratory.
4 Many models in psychology have free parameters that are estimated from the data to which the model is fit. A zero-parameter model is one in which no parameters need to be estimated from the data (all are set a priori).
5 Pronounced "accent."
6 Barring bugs, of course.

REFERENCES

Anderson, J. R. (1976). *Language, Memory, and Thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Anderson, J. R. (1983). *The Architecture of Cognition*. Cambridge, MA: Harvard University Press.
Anderson, J. R. (1990). *The Adaptive Character of Thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Anderson, J. R. (1991). The adaptive nature of human categorization. *Psychological Review*, *98*, 409–429.

Anderson, J. R. (1993). *Rules of the Mind*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Anderson, J. R., Bothell, D., Byrne, M. D. *et al.* (2004). An integrated theory of mind. *Psychological Review*, *11*(4), 1036–1060.
Anderson, J. R., Boyle, C. F., Corbett, A. & Lewis, M. W. (1990). Cognitive modelling and intelligent tutoring. *Artificial Intelligence*, *42*, 7–49.
Anderson, J. R. & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum Associates.
Anderson, J. R. & Milson, R. (1989). Human memory: an adaptive perspective. *Psychological Review*, *96*, 703–719.
Anderson, J. R. & Pirolli, P. (1984). Spread of activation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *10*, 791–798.
Anderson, J. R. & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, *2*, 396–408.
Berners-Lee, T. (1989). *Information Management: A Proposal*. Geneva: CERN.
Blackmon, M. H., Kitajima, M. & Polson, P. G. (2005). Web interactions: tool for accurately predicting website navigation problems, non-problems, problem severity, and effectiveness of repairs. *CHI 2005, ACM Conference on Human Factors in Computing Systems, CHI Letters*, *7*(1), 31–40.
Blackmon, M. H., Polson, P. G., Kitajima, M. & Lewis, C. (2002). Cognitive walkthrough for the web. *CHI 2002, ACM Conference on Human Factors in Computing Systems, CHI Letters*, *4*(1), 463–470.
Brunswik, E. (1952). *The Conceptual Framework of Psychology*. Chicago: University of Chicago Press.
Bush, V. (1945). As we may think. *Atlantic Monthly*, *176*, 101–108.
Byrne, M. D., Anderson, J. R., Douglass, S. & Matessa, M. (1999). Eye tracking the visual search of click-down menus. *CHI 1999, ACM Conference on Human Factors in Computing Systems, CHI Letters*, *1*(1), 402–409.
Card, S. K., Moran, T. P. & Newell, A. (1983). *The Psychology of Human–Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Card, S. K., Pirolli, P., Van Der Wege, M. *et al.* (2001). Information scent as a driver of web behavior graphs: results of a protocol analysis method for web usability. *CHI 2001, ACM Conference on Human Factors in Computing Systems, CHI Letters*, *3*(1), 498–505.
Chi, E. H., Rosien, A., Suppattanasiri, G. *et al.* (2003). The Bloodhound project: automating discovery of web usability issues using the infoscent simulator. *CHI 2003, ACM Conference on Human Factors in Computing Systems, CHI Letters*, *5*(1), 505–512.
Dumais, S. T. (2003). Data-driven approaches to information access. *Cognitive Science*, *27*, 491–524.
Dumais, S. T., Furnas, G. W., Landauer, T. K. *et al.* (1988). Using latent semantic analysis to improve access to textual information. Paper presented at the Conference on Human Factors in Computing Systems, CHI ’88, Washington, DC.
Engelbart, D. C. (1962). *Augmenting Human Intellect: A Conceptual Framework* (No. AFOSR-3223). Menlo Park, CA: Stanford Research Institute.
Furnas, G. W., Landauer, T. K., Gomez, L. W. & Dumais, S. T. (1987). The vocabulary problem in human–system communication. *Communications of the ACM*, *30*, 964–971.
Glimcher, P. W. (2003). *Decisions, Uncertainty, and the Brain: The Science of Neuroeconomics*. Cambridge, MA: MIT Press.
Gray, W. D., John, B. E. & Atwood, M. E. (1993). Project Ernestine: a validation of GOMs for prediction and explanation of real-world task performance. *Human–Computer Interaction*, *8*, 237–309.
Hornof, A. J. & Halverson, T. (2003). Cognitive strategies and eye movements for searching hierarchical computer displays. *CHI 2003, ACM Conference on Human Factors in Computing Systems, CHI Letters*, *5*(1), 249–256.
Hornof, A. J. & Kieras, D. E. (1997). Cognitive modeling reveals menu search is both random and systematic. Paper presented at the Conference on Human Factors in Computing Systems, CHI ’97, New York.

John, B. E., Prevas, K., Salvucci, D. D. & Koedinger, K. (2004). Predictive human performance modeling made easy. *CHI 2003, ACM Conference on Human Factors in Computing Systems, CHI Letters*, *6*(1), 455–462.

Jones, W. P. (1986). The memory extender personal filing system. Paper presented at the Conference Human Factors in Computing System, CHI '86, Boston, MA.

Kieras, D. E. & Meyer, D. E. (1997). An overview of the epic architecture for cognition and performance with application to human-computer interaction. *Human–Computer Interaction*, *00*, 391–438.

Kitajima, M., Blackmon, M. H. & Polson, P. G. (2000). A comprehension-based model of web navigation and its application to web usability analysis. In S. McDonald, Y. Waern & G. Cockton (eds.), *People and Computers XIV, Usability or Else! Proceedings of HCI 2000* (pp. 357–373). Sunderland: Springer-Verlag.

Kitajima, M., Blackmon, M. H. & Polson, P. G. (2005). Cognitive architecture for website design and usability evaluation: Comprehension and information scent in performing by exploration. Paper presented at the Human Computer Interaction International, Las Vegas, NV.

Landauer, T. K. (1986). How much do people remember? Some estimates of the quantity of learned information in long-term memory. *Cognitive Science*, *10*, 477–493.

Landauer, T. K. & Dumais, S. T. (1997). A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, *104*, 211–240.

Lucas, P. (2000). Pervasive information access and the rise of human-information interaction. Paper presented at the Human Factors in Computing Systems, CHI 2000, The Hague.

Luo, L. & John, B. E. (2005). Predicting task execution time on handheld devices using the keystroke-level model. Paper presented at the CHI 2003, ACM Conference on Human Factors in Computing Systems, Late Breaking Results, Portland, OR.

MacKenzie, I. S. (2003). Motor behavior models for human-computer interaction. In J. M. Caroll (ed.), *HCI Models, Theories, and Frameworks* (pp. 27–54). San Francisco, CA: Morgan Kaufman.

Manning, C. D. & Schuetze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Mayr, E. (1983). How to carry out the adaptationist program? *American Naturalist*, *121*, 324–334.

Meyer, D. E. & Kieras, D. E. (1997a). A computational theory of executive cognitive processes and multiple-task performance: I. Basic mechanisms. *Psychological Review*, *104*(1), 3.

Meyer, D. E. & Kieras, D. E. (1997b). A computational theory of executive cognitive processes and multiple-task performance: Part 2. Accounts of psychological refractory-period phenomena. *Psychological Review*, *104*(4), 749.

Neisser, U. (1967). *Cognitive Psychology*. New York: Appleton-Crofts.

Newell, A. (1973). You can't play 20 questions with nature and win: projective comments on the paper of this symposium. In W. G. Chase (ed.), *Visual Information Processing*. New York: Academic Press.

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

Nielsen, J. (2000). *Designing Web Usability*. Indianapolis, IN: New Riders.

Nielsen, J. & Mack, R. L. (eds.) (1974). *Usability Inspection Methods*. New York: John Wiley & Sons.

Norman, D. A. (1998). *The Invisible Computer*. Cambridge, MA: MIT Press.

Pirolli, P. (1997, March). Computational models of information scent-following in a very large browsable text collection. Paper presented at the ACM Conference on Human Factors in Computing Systems, CHI '97, Atlanta, GA.

Pirolli, P. (1999). Cognitive engineering models and cognitive architectures in human-computer interaction. In F. T. Durso, R. S. Nickerson, R. W. Schvaneveldt, S. T. Dumais, D. S. Lindsay & M. T. H. Chi (eds.), *Handbook of Applied Cognition* (pp. 441–477). Chichester: John Wiley & Sons.

Pirolli, P. (2005). Rational analyses of information foraging on the web. *Cognitive Science*, *29*(3), 343–373.

10 Pirolli, P. (in press). *Information Foraging: A Theory of Adaptive Interaction with Information*. Oxford: Oxford University Press.

Pirolli, P. & Card, S. K. (1999). Information foraging. *Psychological Review*, *106*, 643–675.

Pirolli, P. & Fu, W. (2003). SNIF-act: a model of information foraging on the world wide web. In P. Brusilovsky, A. Corbett & F. de Rosis (eds.), *User Modeling 2003, 9th International Conference, UM 2003* (Vol. 2702, pp. 45–54). Johnstown, PA: Springer-Verlag.

Salvucci, D. D. (2001). An integrated model of eye movements and visual encoding. *Cognitive Systems Research*, *1*, 201–220.

Singley, M. K. & Anderson, J. R. (1989). *Transfer of Cognitive Skill*. Cambridge, MA: Harvard University Press.

Spool, J. M., Perfetti, C. & Brittan, D. (2004). *Designing for the Scent of Information*. Middleton, MA: User Interface Engineering.

Spool, J. M., Scanlon, T., Schroeder, W. *et al*. (1999). *Website Usability*. San Francisco, CA: Morgan Kaufman.

Stephens, D. W. & Krebs, J. R. (1986). *Foraging Theory*. Princeton, NJ: Princeton University Press.

Tinbergen, N. (1963). On the aims and methods of ethology. *Zeitschrift für Tierpsychologie*, *20*, 410–463.

Treisman, A. M. & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, *12*, 97–136.

Turney, P. D. (2001, September). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. Paper presented at the Twelfth European Conference on Machine Learning, ECML 2001, Freiburg, Germany.

Winterhalder, B. & Smith, E. A. (1992). Evolutionary ecology and the social sciences. In E. A. Smith & B. Winterhalder (eds.), *Evolutionary Ecology and Human Behavior* (pp. 3–23). New York: de Gruyter.

AUTHOR QUERY-FORM

Dear Author,

During the preparation of your manuscript for publication, the questions listed below have arisen. Please attend to these matters and return this form with your proof.

Many thanks for your assistance.

Query References	Query	Remarks
1.	Au: **Miller (1956)** has not been found in the references.	
2.	Au: **Simon (1974)** has not been found in the references.	
3.	Au: **John & Kieras (1996a)** has not been found in the references.	
4.	Au: **Miller (1956)** has not been found in the references.	
5.	Au: **Ryle (1949)** has not been found in the references.	
6.	Au: **Miller (1956)** has not been found in the references.	
7.	Au: **Simon (1974)** has not been found in the references.	
8.	Au: Please update the year for this citation.	
9.	Au: Please supply the volume number.	
10.	Au: Please update the year of publication in this reference.	
11.	Au: **Newell, 1983** has been changed to **Newell, 1990** so that this citation matches the reference list. Please confirm that this is correct.	