Cyberpsychology: An Introduction to Human–Computer Interaction

Kent L. Norman

University of Maryland
Learning and Memory, Transfer and Interference

Scenario 1

Mary loves to do everything online, such as shopping, e-mail, and banking. She figures that she has about thirty different usernames and passwords for her various accounts and memberships. If she had her druthers, she would use the same username and password for all of them, but she knows that that is a bad idea for security reasons. Furthermore, some systems automatically assign her a username and/or password that cannot be changed, and for others, her desired username was already taken or her password did not meet the security requirements. So, she is stuck with thirty usernames and passwords. She tried in vain to memorize them. In fact, she made up a table that includes the system name or Web site, the username, and the password so that she could quiz herself. The harder she tried, the worse it got. She finally gave up and decided to put them all on her PDA and lock the file with one password.

Scenario 2

John is a retired computer programmer. He started programming in the late 1960s using the language FORTRAN. He took courses to learn the language and studied the manuals and documentation. He knew the syntax of the language perfectly. He had memorized the names of all of the functions he might use. He knew the structure of the language and had even memorized many of the algorithms he used for sorting, accessing, and processing information. He worked for 25 years programming applications for his company. Then his boss announced that they were switching to a new language, C++. He started taking courses in C++ but was having a terrible time. The younger students
who had little or no programming experience were doing better than he was. John decided to take early retirement.

Overview

This chapter deals with learning and memory issues on the part of the human and the computer. What must the human learn and remember about the computer (e.g., commands, procedures, passwords)? What does the computer learn and remember about the human and the environment (e.g., preferences, logs, knowledge)? How can the human use the computer to retrieve information? How does the computer get help from the user to retrieve information? When it comes to information storage and retrieval, a synergistic relationship is required between the human and the computer.

We see that for both humans and computers, memory requires three processes: encoding, storage, and retrieval. The information must first be represented in an appropriate manner; that is, it must be processed into an appropriate code. Second, it must be stored in some format and media in a memory location. Finally, it must be retrieved using some cue or address and decoded to an output form. Beyond this, as noted in Chapter 2, the differences between human memory and computer memory are immense. Moreover, there are a number of factors that make learning and memory difficult for humans. Scenario 1 illustrates one of many human memory problems caused by computer technology and a possible solution. Scenario 2 illustrates the problem of relearning and negative transfer due to changes in software and technology over time.

Learning about Computers and the Human–Computer Interface

What one learns about computers and the human–computer interface comes from both formal and informal learning. You may have taken courses about computers, information technology, and various applications. You may have read books and manuals and gone through tutorials on word processors, spreadsheets, and OSs. You may have explored computers and programs on your own. You may have had many hours of experience working with computers, programs, and browsing the Web. You may even be learning more by reading this book.

Much of what you know about computers is “head knowledge,” but you have also learned a lot of mousing and keyboarding skills; attitudes, preferences, and feelings about computers; and expectations, associations, and intuitions about using computers. We make a distinction among various types of knowledge. Declarative knowledge is about facts, events, and objects. Memories are either explicit or implicit. Explicit memories are those that we can consciously recollect (e.g., your password, the name of your program,
the menu item to resize an object, how much storage you have on your hard drive). **Implicit memories affect our thoughts without conscious recollection.** Implicit memories may be about feelings. One may have anxieties about computers from early traumatic experiences (e.g., being yelled at as a child for accidentally erasing an important file). Implicit memories also include many things that are familiar to you but that you cannot explicitly recall. For example, it is very unlikely that you can recall the pull-down menu items in a word processor or an OS that you use all of the time. It is unlikely that you can list the letters in order on the keyboard, even though you are a fast touch typist. Finally, implicit memories influence what we think. Thinking about sports can implicitly affect or prime subsequent thoughts and behaviors.

Declarative knowledge in memory is composed of general or **semantic memory** about things independent of time and place (e.g., "#FFFFFF" is the code for white, "EXIT" is used to quit a program, binary code is composed of 1s and 0s) and **episodic memory** about factual information that is acquired at a specific time and place (e.g., my hard disk crashed last week, my new computer was delivered to my home on Tuesday).

Finally, **procedural memory** contains information about how to perform skilled motor activities, such as driving a car, typing, and drawing a cartoon. Procedural memory includes not only the sequencing of actions, but also the fine-tuning of motor movements. Today, computers with GUIs require a considerable amount of procedural memory to perform most tasks. If you want to watch a video on your screen, you need to know the sequence of steps to open the video player, select the clip, control the functions, and adjust parameters such as the volume. The same is true for most electronic technology from video games to MP3 players and from digital cameras to cell phones. One needs to learn the sequence of both discrete steps and continuous adjustments.

**Verbal Learning and Memory**

There are many forms of human learning and memory, and as seen in the previous section, the human–computer interface impacts them all. In general, learning is the acquisition of knowledge. Knowledge may be as simple as an association between two stimuli, such as a bell and food in the classical conditioning paradigm studied by Pavlov (1927), or between an action and a response, such as a key press and the delivery of a gum ball studied by Skinner (1953), or it may be as complex as learning a language or learning to play a violin. **Unlike animal learning, human learning and memory are often verbal.**

Human memory was first studied systematically by Herman Ebbinghaus (1850–1909). He wanted to study the rate of acquisition of different types of verbal memory and the process of forgetting. He realized, however, that in using common words (e.g., dog, house, tree), there were already many associations among the words that could obscure the effects he wanted to study. His solution was to use nonsense words (e.g., VUTAW, ZOREX, MINIW), syllables, and letter combinations, such as “trigrams” (e.g., XOR, VIH, BEV), that provided little or no meaning to the subjects.
Table 6.1. Paired Associate Problem of Usernames and Passwords

<table>
<thead>
<tr>
<th>System/Web site</th>
<th>Username</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home computer</td>
<td>mike</td>
<td>secret</td>
</tr>
<tr>
<td>Office computer</td>
<td>mike</td>
<td>2203LIB</td>
</tr>
<tr>
<td>Office network</td>
<td>msmith</td>
<td>2203LIB</td>
</tr>
<tr>
<td>Hotmail.com</td>
<td>Msmith45</td>
<td>Mail4me</td>
</tr>
<tr>
<td>Amazon.com</td>
<td><a href="mailto:msmith@hotmail.com">msmith@hotmail.com</a></td>
<td>Books4me</td>
</tr>
<tr>
<td>USA.com</td>
<td>msmith45</td>
<td>B34h456</td>
</tr>
<tr>
<td>WashingtonPost.com</td>
<td>msmith1965</td>
<td>Paper2day</td>
</tr>
<tr>
<td>Cappella.edu</td>
<td><a href="mailto:Msmith1965@gmail.com">Msmith1965@gmail.com</a></td>
<td>3546574</td>
</tr>
<tr>
<td>Firstusa.com</td>
<td>Msmith34</td>
<td>D7GH8J45</td>
</tr>
<tr>
<td>Wikipedia.com</td>
<td><a href="mailto:Msmith1965@gmail.com">Msmith1965@gmail.com</a></td>
<td>Mork34</td>
</tr>
<tr>
<td>eBay.com</td>
<td>Msmith1965</td>
<td>Mork34</td>
</tr>
<tr>
<td>Wordpress.com</td>
<td>Mike1965</td>
<td>myblog</td>
</tr>
<tr>
<td>Cingular.com</td>
<td>Mikesmith36</td>
<td>222656</td>
</tr>
<tr>
<td>Networksolutions.com</td>
<td>msmith</td>
<td>Mork34</td>
</tr>
</tbody>
</table>

It is important to realize that no learning occurs in the absence of prior knowledge, even at birth. When we learn something new, there is already some prior learning that has taken place. Others have tried to control the amount of prior knowledge by using items that were matched on familiarity or frequency of use in the language. The same problem exists in learning things at the human–computer interface. Terms such as “file,” “tools,” “style,” and “repaginate” have different levels of familiarity and meaningfulness to the user.

Ebbinghaus developed a number of paradigms for studying memory, which are still in use today and which characterize the types of learning embedded in many aspects of the human–computer interface.

Paired Associate Learning

In a paired associate learning task, you must learn associations between pairs of words. Given Word A (stimulus), you must learn to say Word B (response). We might have sixteen pairs of words that we present in a random order on each trial. On the first trial, we present the first word in the pair and then the two words together. On the second trial, we present the first word in the pair, the subject has to say the second word before the time is up (say, 3 s), and then the pair is shown together. Over a number of trials, the subject will learn to recall all response words.

Paired associate learning occurs in many situations, such as when learning vocabulary in a foreign language (e.g., “window”–“Fenster”), when associating events and dates (e.g., Balfour Declaration–1917), and when associating works of art and artists (e.g., Mona Lisa–Leonardo da Vinci). In HCI, there are also many cases of paired associate learning. Table 6.1 shows the problem of learning usernames and passwords for different systems and Web sites. Additional paired associate learning occurs when associating functions and
their names (e.g., find the square root of X–Math.sqrt(X)) and menu options and their shortcuts (e.g., paste–Ctrl-V).

Research on learning paired associate lists has shown that unfamiliar words are harder to learn than familiar words. Longer lists are harder than shorter lists. It takes more trials to learn the list with short study time intervals than with long study intervals. If the stimulus words are used as responses for other stimulus words or the response for one stimulus is also the response for another stimulus, confusion can occur, and it will be harder to learn the list. These findings have direct implications on HCI. It is easy to see why some systems are very hard to learn.

**Serial List Learning**

In a serial list learning task, you are to learn a list of words in order. On the first learning trial, you see each word one at a time for a fixed time interval. On the second trial, you are shown a prompt; then you are to say the first word on the list; after the time is up, you are shown the first word; then you are to say the second word before the time is up, and so on to the end of the list. Serial lists occur in many contexts—learning the colors in the rainbow, the months in a year, and the digits in a phone number.

In HCI, **serial lists are embedded in procedures and in sequences of menu selections to perform tasks** (e.g., to send an e-mail: OPEN mail, select FILE, select NEW MESSAGE, enter address, subject, and body of e-mail, select SEND).

A curious thing that Ebbinghaus discovered with serial lists is that the probability of recalling the item depends on its position in the list. As shown in Figure 6.1, there is a higher probability of recalling early items (**primacy effect**) and of recalling the last items (**recency effect**). Items in the middle are subject to more interference and confusion. Moreover, the longer the list, the worse it is, particularly for the items in the middle. Consequently, computer procedures that involve many steps are hard to learn, and the greatest problems in recall will be in the middle of the procedures.
Free Recall

In a free recall learning task, you are shown a list of words during a study time and then asked to recall them in any order during the test period. There may be repeated study and test trials. It is typically found that the more familiar the words, the easier they are to recall, and the longer the list, the harder it is to recall all of the words. However, the more associations there are among the words, the easier it is to recall the words. It is also found that the words are not always recalled in the order in which they are presented or in a random order but rather clustered according to meaningful groups and associations. Yet, it is often the case that the last words in the list are recalled first.

Free recall situations are frequent in human–computer interfaces. We are required to know lists of commands, options, categories, files, Web sites, etc. See how many top-level URL domain names (e.g., .com) you can recall other than the country names (e.g., .us, .cn). Fortunately, for many lists we do not have to recall the items but rather recognize the one we want.

Recognition Tasks

There are several types of recognition tasks. In the old-new task, you are shown a series of words or pictures. For each item, you are asked to indicate whether the word or image appeared previously on the list. Because this is a “yes–no” detection task, it is analogous to the signal detection task that is discussed in Chapter 5 and can be analyzed in the same way. Recognition is easy if the items are unique and distinct, but if similar, confusable items are used, it can be very difficult.

The second type of recognition task is the correct-incorrect task. Instead of recalling the word in a paired associate task, you are shown a word and asked whether it is the correct or incorrect associate to the word. This is like a typical “true–false” question on an exam. Finally, the forced-choice recognition task presents the stimulus word with a list of possible associates from which you are to pick. This is like a typical “multiple-choice” item on an exam. Again, recognition is difficult to the extent that items are similar and confusable.

These types of recognition tasks are prevalent in GUIs. The advantage is that they do not require the recall of commands, but rather the selection of menu options. Recognition of the correct option often depends on its distinctiveness from other options in the set (Schwartz & Norman, 1986).

Learning Curves

Acquisition of information by humans is not as immediate and complete as it usually is with computers. Instead, it generally requires time and repeated exposures to the information. Ebbinghaus found that with repetitions of
pairings or exposure to lists and the amount of study time, the probability of recall increases. The function relating learning trials or study time to performance is called a learning curve. Learning curves can be of several different shapes, depending on the type and difficulty of the material, as shown in Figure 6.2. All learning curves have an initial starting point determined by the chance of guessing and amount of prior knowledge, as well as a final ending point called the asymptote. For some tasks, the asymptote may be at 100 percent, but for many realistic tasks, few people reach perfect performance.

If each trial adds a constant amount to what has already been learned and nothing already learned is forgotten, the learning curve will be linear and will plot as a straight line until maximum performance is reached. Thus, each trial adds the same amount until everything is learned. Somewhat surprisingly, linear learning curves are rarely seen in theory or in practice. Instead, we find exponential curves in situations where easy items in the set are learned quickly at the beginning, causing a fast startup, but then it takes many more trials to learn the hard items, as shown in Figure 6.2. Many user applications are designed to be quickly learned at the beginning, but then to become proficient in all functions requires a lot of time. Most e-mail, word processor, and spreadsheet programs are learned this way. However, for other applications, and particularly for programming languages, learning can be difficult and slow at the beginning because many new, unfamiliar things must be learned before you can really get going. This results in what looks like an S-shaped learning curve. Performance increases slowly at the beginning, followed by a more rapid acquisition of items, and then ending with a slower rate to reach asymptote.

The term “steep learning curve” has become a misnomer. Technically, if performance is plotted as a function of study trials, the exponential learning curve in Figure 6.2 would be described as steep. However, in current parlance, a steep learning curve has come to refer to a very difficult learning situation. The metaphor is that it takes a lot more work to climb a steep slope than a gradual incline. In this latter sense, the curve reverses the axes of the graph and plots the number of learning trials that it takes on the y-axis to achieve a certain level of performance on the x-axis.
Forgetting Curve

In addition to the acquisition of information, Ebbinghaus also studied forgetting. Forgetting is a very human aspect of memory that can be due to several factors, some similar to computers and others quite different. The deletion of information in a computer is generally directed by a command to delete and erase files or by a computer malfunction. Human memory, in contrast, may decay over time, or recall may be inhibited by other interfering information. In general, if there is no additional study following learning, the probability of recall drops exponentially, as shown in Figure 6.3.

The good news is that although a lot of information is lost quickly over time, the curve does not go to zero. Some portion of the information is retained, and if there is a relearning phase, acquisition of forgotten material is faster than if it had not been learned previously. Moreover, the rate of forgetting following relearning is less than that following original learning, hence the importance of refresher classes.

Three Memory Stores

Even from early studies of memory, it was believed that memory was not a single unit or system. William James proposed that there were primary and secondary systems one that was for lasting memory and the other for temporary memory. Contemporary research suggests that it is divided into three systems or stages, as shown in Figure 6.4, with a sensory register, a short-term memory, and a long-term memory with an executive processor in charge (Atkinson & Shiffrin, 1968). This three-stage model helps explain many of the phenomena that we find in memory research and is an important determinant in how we learn and remember things at the human–computer interface.
Sensory Register

When you glance at a screen, your visual system takes in an amazing amount of information. If the screen goes blank or is flipped to another screen, the information you saw remains in your visual system for only a moment. You have only a short time to attend to the information and to glean what you can from that image before it is gone, unless you are an unusual individual with eidetic imagery, or a photographic memory. The question is how much information you can acquire in a glance and pass on to short-term memory.

Information in the short-term sensory register is rich in its original sensory form, but decays rapidly as information is extracted for further processing (Rainer & Miller, 2002). It is probably more accurate to think of multiple sensory memories. Iconic memory is the name of the visual sensory memory. This information is like a snapshot that decays within one-fourth of a second. Echoic memory is for the auditory sense. Information for this sense is retained for several seconds. Much less is known about the sensory memories for smell and touch.

Imagine again being shown a Web page for only a split second. How much information could you extract from the visual image? Even though iconic memory is extremely rich, studies by Sperling (1960) suggest that you will only be able to recall three to five items. In Sperling's classic experiment, he presented $3 \times 3$ arrays of random letters for approximately one-twentieth of a second; subjects could only recall four or five of the letters. Sperling hypothesized that even though the letters were in iconic memory, subjects did not have the time to process and name the letters before the memory...
Learning and Memory, Transfer and Interference • 159

had decayed. To test this theory, he provided a cue immediately after the presentation to tell the subjects to recall only the letters in the first, second, or third row of the array using a high, medium, or low tone. With this procedure, subjects were able to recall nearly all letters in a row and, by extrapolation, would have recalled many more letters if they had had time before the image had decayed.

The beauty of many computer interfaces is that if the image disappears before you can extract the needed information, you can return to the screen or scan back to previous frames in a video. As we explore later, computer memory can bolster or make up for the deficits of our short-term sensory store and help us extract the information that we need.

**Short-Term Memory**

Short-term memory is a buffer in which we hold units of information for a limited time that we have extracted from the sensory register. Short-term memory system has a limited capacity. George A. Miller (1956) in his classic article, “The Magical Number Seven, Plus or Minus Two,” specifies the limit as $7 \pm 2$. The best example is the test of memory span for digits. In this test, a list of digits is read to a subject who then has to repeat the numbers back in the same order without making a mistake. On average, most college students can remember lists of eight or nine digits without making errors. Short-term memory is constantly in action at the human-computer interface. We are trying to remember names, numbers, settings, pages, and steps as we interact with machines. We keep track of what is on the clipboard, what files are open, etc.

Information is lost from short-term memory due to decay over time, interference among items, and being displaced by incoming information. In other words, if we are trying to juggle too many items, some of them get dropped.

Information in short-term memory can actually be retained for long periods of time using **maintenance rehearsal**—that is, by repeating it over and over to oneself to refresh the items. Maintenance rehearsal seems to be accomplished primarily through inner vocalization and requires considerable cognitive effort and concentration. If you are distracted, the information is lost. Repeating items to yourself does little to transfer the information to long-term memory. In contrast, **elaborative rehearsal** serves this purpose. Items are elaborated by adding meaning to them (“sense making”) and relating them to other information in long-term memory.

Although the capacity of short-term memory is limited to $7 \pm 2$ items, the items can be large chunks of information. So, for example, you may be trying to remember the password “E6HJ9K2” with seven characters, or you may be trying to remember the seven words “WALNUT, EAGLE, STARFISH, CORN, ROCK, TUNA, MOUNTAIN” made up of many more characters. Short-term memory is limited by the number of chunks, not the size of the
chunks. This means that if we can convert strings of information into meaningful chunks, we can retain much more information in short-term memory.

In the past, short-term memory had been thought of primarily for storage and transfer of information to long-term memory. To account for how people perform cognitive tasks such as problem solving and decision making, Alan Baddeley (1998, 2000) and others have proposed a more complex system that involves working memory. This memory system serves as a workbench on which information is manipulated and assembled to perform cognitive tasks. Although the distinction between short-term and working memory is not entirely clear, the idea is that working memory is a dynamic workspace rather than merely a short-term storage area for information on its way to long-term memory (Nyberg, Forkstam, Petersson, Cabeza, & Ingvar, 2002).

Working memory is controlled by a central executive and has at least two storage areas, a phonological loop and a visuospatial working memory. The phonological loop allows for the rehearsal of speech-based information. Words can be temporarily stored and manipulated in the loop. The visuospatial memory stores visual images and spatial information. It is like a scratch pad in that images can be temporarily stored and manipulated. It appears that the phonological loop and the visuospatial memory function relatively independently so that you can rehearse numbers in one while manipulating the arrangement of letters in the other (Baddeley & Hitch, 1974; Reed, 2001). The central executive controls information in and between the phonological loop, visuospatial memory, and long-term memory. The central executive controls attention, does planning, selects strategies for processing information, and monitors the progress of cognitive activities.

In HCI, the central executive controls attention, what we look at, and what we extract from the image. It controls what goes into working memory (e.g., a list of menu options) and decision strategies for selecting the appropriate item. It helps keep track of the spatial organization of windows and icons that are not currently visible but are being thought about. How the central executive does this is not clear. However, the idea of such an executive emphasizes the need for control over the memory processes in support of thinking and problem solving. We return to this issue in Chapter 7 and later in the context of AI.

Long-Term Memory

Long-term memory, the relatively permanent collection of information and knowledge that we all acquire, is truly remarkable. We may complain about our memories for facts and events, but given the sheer vastness of long-term memory, we should be more amazed than dismayed. John von Neumann, a mathematician and computer scientist, estimated the size of human long-term memory to be $2.8 \times 10^{20}$ bits of information, given no forgetting. My office computer has 280 GB of hard disk storage. This would equate the size of human memory to 1 billion of these computers. Such magnitudes of
computer memory at this point in time can only be achieved at a global level with large server farms on vast networks.

Long-term memory is complex. As discussed previously, it includes both explicit and implicit memory. Explicit memory includes semantic and episodic memory, and implicit memory includes procedural, priming, and classical conditioning memory. Moreover, long-term memory involves a multimodal coding of information to include the following types of information:

- **Spatial/visual information**: pictures, images, symbols, spatial structures, cognitive maps
- **Law and properties**: physical, social, and behavioral laws; properties of things
- **Beliefs, values, and attitudes**: subjective feelings about things, preferences
- **Procedural skills**: plans for how to do things, motor skills
- **Perceptual skills**: interpreting and understanding sensory input

Information in the long-term store is obviously more permanent than short-term memory, but sometimes the information can be hard to retrieve. We may know that it is there or experience the “tip of the tongue” phenomenon. Given enough time, it may be spontaneously remembered. Or given enough searching through cues and associations, it may be recalled. Remembering may be a passive phenomenon in which memories just surface effortlessly, or it may be an active process of digging and problem solving. When we try to recall some information—such as “What were you doing at 3:00 in the afternoon on June 30, 2 years ago?” or “What is the algorithm for a bubble sort?”—we have to actively retrieve it and, in some cases, fabricate what we cannot remember from the things that we can remember. Recall in Chapter 4 the retrospective procedure in usability testing. Users have to recall what they were thinking about while performing a task. One criticism of the procedure is that participants may not actually remember, but instead make up a rational explanation from what they do remember.

Long-term memory is organized. This is evident from the fact that the order of recall of items is neither random nor in the same order as they were learned. Instead, recall is a function of the structure of knowledge and the relationships among items. Human memory is dynamically self-organizing, sometimes consciously and sometimes unconsciously, perhaps even while we are asleep. Computer memory may or may not be organized, depending on how it was stored in files and databases. The organization of human memory is of great interest to computer scientists as a model for computer memory. Similarly, computer models of memory are of great interest to cognitive psychologists for developing models of human memory.

**Organization of Long-Term Memory**

There are four major theories for how long-term memory is organized: hierarchies, semantic networks, schemas and scripts, and connectionist networks.
These are not mutually exclusive but may each operate in different areas and at different times.

Hierarchies

A hierarchy is an organization from general to specific categories. Much of our knowledge about the world is hierarchical in nature. The animal kingdom and the plant kingdom are hierarchical by class, genus, and species. Genealogies are hierarchical family trees. Organization charts of businesses and government agencies are hierarchical. We tend to organize knowledge in hierarchical indexes and outlines. Most computer files are also organized hierarchically with directories, subdirectories, and so on.

Evidence for hierarchical organization of information in long-term memory comes from two basic lines of research. Early research by Bower, Clark, Winzenz, and Lesgold (1969) showed that participants who were presented words in hierarchies recalled them better than those who were presented words in random groups. Others have found that recall is superior when we organize the to-be-learned information hierarchically (Bruning, Schraw, & Ronning, 1999). This suggests that human–computer interfaces that are often inherently hierarchical should be presented and learned in a hierarchical manner. In a very early study on learning options in hierarchical menu selection systems, Parton, Huffman, Pridgen, Norman, and Shneiderman (1985) found that having participants study the hierarchy was superior to any other method of learning.

A second line of research suggests that when we search long-term memory to answer questions, the time that it takes to answer the question is a function of the position of the information in the hierarchy. Collins and Quillian (1969) found that it takes longer to answer the question “Is a canary a bird?” than “Is a canary yellow?” They theorize that our knowledge about animals is stored in a hierarchy with unique information about animals at the first level and general information at second and third levels in the hierarchy. Moreover, it takes a certain amount of time to travel from one node in the hierarchy to another. The more nodes that one must visit, the longer it takes, just as it does if you were trying to locate a particular file in a directory structure on a computer.

Semantic Networks

Strict hierarchies in memory are overly restrictive and do not account for a number of empirical findings. For example, it takes longer to answer true–false to “A dog is a mammal” than it takes for “A dog is an animal,” even though the second statement addresses information that is further up the hierarchy. Moreover, there are large differences in times at the same level. For example, it takes longer to answer true–false for “An ostrich is a bird” than
it does for “A canary is a bird.” Another approach is to think of long-term memory as a **multidimensional network of semantic associations between items.** To answer questions, we again search through this network, but now distances between any two associated items represent their strength of association and affect the time it takes to traverse the network. In a sense, it is like the telephone network or the Internet with trunk lines, routers, and transmission lines. The difference is that neural transmission is very slow in the brain, whereas transmission is near the speed of light in electronic networks.

*Spreading activation theory* was proposed by Collins and Loftus (1975), who assume that there is a complex association network in which memories are distributed in a conceptual space. Links represent associations between nodes. Long links indicate remote associations, and short links close associations. When a memory is activated, its activation spreads outward. This may have the affect of priming related memories in the network and could explain why we respond to different thoughts and stimuli the way we do.

We know from research on networks and the storage of information in networked databases that some systems are more efficient than others. Organization, indexing, and sorting of information facilitates the search and recall of the information at a later time. Google is more successful than other search engines because it uses a more efficient algorithm that results in faster search times. One wonders if we, as humans, might learn more effective strategies of storing and indexing information in our long-term memories.

### Schemas and Scripts

Much of our knowledge has to do with general **structures and frameworks (schemas)** and with **stories and sequences of events (scripts).** A schema is a mental concept or mental model (see Chapter 3) that helps interpret and organize new information. Schemas are ways that we think about things (Jou, Shanteau, & Harris, 1996). We have schemas for celebrations such as weddings, graduations, and birthday parties; for working on computers including such tasks as writing papers, downloading files, and troubleshooting; and for how systems work such as wireless networks, UNIX, and high-definition televisions. Scripts are schemas that pertain to sequences of events (Schank & Abelson, 1977). They are typically procedures and story lines that contain placeholders for characters and props. We have scripts for eating in a restaurant, sending e-mail, and shopping online and offline.

Schemas and scripts differ from semantic networks in that the structure of the links themselves contains information rather than just the nodes. In contrast to semantic networks, where tens of thousands of words are associated in semantic networks, schema theory organizes information into general structures, patterns, and forms along the lines of Gestalt theory. The efficiency of one’s long-term memory then is a function of the quality and usefulness of the schemas that have been stored and used to interpret new information coming in through elaborative rehearsal.
Connectionist Networks

The previous theories of long-term memory are high on conceptualization but low on implementation. The question is "How is long-term memory actually stored in the neural network of the brain?" Connectionist theorists propose that memory is stored throughout the brain in connections among the neurons (Dehaene & Naccache, 2001; Humphreys, Tehan, O'Shea, & Bolland, 2000). Rather than focusing on the abstractions of memory, these theorists build computer simulations of neural networks that attempt to mimic the properties of the neurons exciting or inhibiting nodes across synaptic connections in a network. Numerous simulations have been successful in predicting the patterns of results in memory experiments (Marcus, 2001; McClelland & Rumelhart, 1986). The belief is that, given a large enough network with appropriate complexity, the computer can simulate learning and memory of the brain. In fact, some believe that neural network models may be the basis for a new generation of AI, as we see in Chapter 13. However, an important criticism of connectionist theories is that, given enough complexity in the simulation, you can fit any set of data and still not understand the process.

Levels of Processing

The previously discussed three-stage model of memory is very appealing; however, it is undoubtedly an oversimplification of the human memory system. Another approach to understanding how human memory works is to combine memory with the processing or encoding of information. We do not store the undigested image of the screen. Instead, we process it in various ways, extracting different types of information over time according to our particular needs.

In their levels of processing theory, Craik and Lockhart (1972) proposed an encoding process that governs memory. We first have to attend to the information before we can encode it. Then we process the information along a continuum from shallow to deep. At the shallow level, we encode the sensory or physical features of the stimuli. We detect lines, angles, and contours of letters or objects, or we detect the frequency, tone, loudness, and duration of sounds. If we stop here, little or nothing is stored or remembered. At the intermediate level, we encode names of letters, whole words, and objects. They may be remembered, but because they are disconnected, they may not be retrieved. At the deepest level, information is given semantic meaning with associations to other memories. The more associations, the deeper the processing (Lee, Cheung, & Wurm, 2000; Otten, Henson, & Rugg, 2001).

The levels of processing theory help explain why we can work at some tasks and remember almost nothing about them. For example, you can go through a document and find all underlined words and convert them to italics and not remember what the words were. Or you can even delete e-mails based on how suspicious you are of their subject lines and not remember any of the words in the titles.
According to the levels of processing theory, the more we process and elaborate the information, the more likely we are to recall it (Craik & Tulving, 1975). So, for example, if you are learning a set of menu options (e.g., artistic, blur, brush strokes, distort, pixilate, sharpen, sketch, stylize, texture, video) in an application such as Adobe Illustrator, making them more meaningful with explanations, seeing examples, and relating them to your own work will help elaborate them.

Recognition, Recall, and Retrieval

Menu selection and GUls rely on recognition memory. You do not have to recall the names of the functions or options, you just have to recognize them from a list. If you want to delete a file, you recognize the file icon, recognize the trash icon, and drag the file icon to the trash icon. Command-line and programming language systems require the recall of terms from long-term memory. To delete the file, you have to recall the name of the file, the name of the function to delete a file, and type the command using the correct syntax.

Recall of information from memory requires a retrieval process that generally begins with a cue or a question. What is the name of the Web site for online auctions? If you cannot remember, it is often useful to generate additional cues (Allan, Wolf, Rosenthal, & Rugg, 2001; Halpern, 1996). Try going through the alphabet to see if it helps retrieval. Think of images, screen shots, and logos.

When items are stored in long-term memory, they are associated with the cues present at the time of storage. The encoding specificity principle says that information present at the time of encoding can serve as an effective cue for later retrieval (Hannon & Craik, 2001; Tulving & Thomson, 1973). Part of this information can be the context – when and where – it was learned. The principle of context-dependent memory says that information is better recalled in the same context in which it was learned than in a different context (Smith & Vela, 2001). That is, if you learn something while sitting in front of your computer, it will be recalled better when you are in front of your computer than when you are working under your car. Something that you read in a book in the park will be recalled better when you are in the park than when you are taking an online test for your driver’s license. Knowing this, it is best either to study in the same context as the test or to at least imagine that context while studying.

Another part of the information at the point of encoding has to do with your internal state. The principle of state-dependent memory says that we are more likely to remember information when our psychological state or mood is similar at both encoding and retrieval (Weissenborn & Duka, 2000). Moreover, when we are in a happy mood, we are more likely to remember positive experiences, and when we are in a negative mood, we are more likely to remember negative experiences (Mineka & Nugent, 1995). Obviously, as we interact with computers, our moods can be random, and this can affect...
Table 6.2. Interference Paradigms

<table>
<thead>
<tr>
<th></th>
<th>Task 1</th>
<th>Task 2</th>
<th>Interval</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proactive Interference</td>
<td>Learn</td>
<td>Learn</td>
<td></td>
<td>Recall</td>
</tr>
<tr>
<td>Experimental group</td>
<td>List A</td>
<td>List B</td>
<td></td>
<td>List B</td>
</tr>
<tr>
<td>Control group</td>
<td>Learn</td>
<td>List B</td>
<td></td>
<td>Recall</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retroactive Interference</td>
<td>Learn</td>
<td>Learn</td>
<td></td>
<td>Recall</td>
</tr>
<tr>
<td>Experimental group</td>
<td>List A</td>
<td>List B</td>
<td></td>
<td>List A</td>
</tr>
<tr>
<td>Control group</td>
<td>Learn</td>
<td>List B</td>
<td></td>
<td>Recall</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

what we recall about the content of e-mail, online articles, and successful or frustrating things about the interaction.

Transfer of Training and Interference

In Scenario 2 at the beginning of the chapter, John worked for years using FORTRAN until his company implemented C++. When he began taking courses to learn C++, he believed that his programming experience would help him do better than the novices in the class. What went wrong? To the extent that the information in Task A is similar to Task B, there will be positive transfer; to the extent that they are different, there will be less transfer; and to the extent that the information is contradictory, there can be negative transfer. For John, there was probably some positive transfer of general concepts, but there was a lot that was contradictory and that led to negative transfer.

We distinguish between two types of interference, depending on the sequence of learning and recall situations, as shown in Table 6.2. **Proactive interference** occurs when something you learned previously interferes with something you learned later that you are trying to recall. This is John’s problem. As computers have evolved, programming languages, OSs, and applications have changed substantially. People who have learned a number of languages, systems, and versions over time have built up a repository of interference with learning newer things. This becomes a significant problem with age in an environment of changing technology. The more you learned in the past that is now obsolete, the worse off you will be in the future. Even minor changes in the names of functions, menus, and icons from one version of an application to another can cause problems with proactive interference and lead to intrusion errors from the old version.

**Retroactive interference** occurs when you learn one thing and then a second, and then try to recall the first thing that you learned. It especially becomes a problem when you are trying to return to an older system or version after having learned a newer one. Perhaps this is not a frequent problem,
but it is still likely to occur in a changing environment when one needs to recover something from the past such as a legacy system.

When designing new versions of systems or new training methods, or analyzing the effects of change, one must consider the types of things that transfer. Table 6.3 lists various types of transfer that one should take into consideration. Consider a person who has only used MS Windows XP for years and is now shifting to Apple Mac OS X. According to Table 6.3, this type of transfer might be characterized as far, mixed valence, horizontal, figural, high road, and backward reaching. Whereas, a person who is moving from MS Windows XP to MS Windows Vista would anticipate a transfer that would be near, positive, horizontal, literal, low road, and forward reaching.

### External Learning and Memory

Research on metamemory and metacognition looks at our ability to judge how difficult it will be to learn something and whether we will be able to recognize or recall it later (Metcalfe & Shimamura, 1996; Nelson, 1992). In Scenario 1, Mary realized that learning and recalling thirty different passwords would be a problem. Her solution was to offload the information onto her PDA and lock it with one password.

When it comes to storing and retrieving information, the computer has an easy time of it compared to humans. This is because learning and memory in a computer are fundamentally different from those functions in a human (see Chapter 2). We do not discuss this further in this section, but instead think about the ways that computers can store and use information in the process of interacting with humans.
We have many strategies for supplementing our fragile and volatile human memory. We have always used mnemonics to aid internal memory and written notes as external memory aids, but current digital technologies have added many new possibilities. Have you ever wanted a video recording of your day so that you could go back and replay a contentious conversation with a colleague or find a misplaced library book? Although it is technically possible to do this, it is not yet very practical.

The main questions are “How can we use technology to improve, enhance, or supplement our memory?” and “When should we offload that memory to the computer?” Related questions are as follows:

- Should I memorize telephone numbers, prices, part numbers, and so on, or store them in external memory on a computer, PDA, or other device?
- What should be left to human memory, and what should be put on the computer?
- What is the best way to input data into the computer, and how is it accessed later?

In Figure 6.4, Atkinson and Shiffrin (1968) included an executive processor that monitors memory storage and decides whether to memorize and store information in long-term memory, and Baddeley (1993) introduced the idea of a working memory to store information during processing. To these concepts, we must also add additional executive processor functions to offload and store information on external devices and memorize methods of retrieval.

Personal information management systems (PIMSs) are to applications specifically written for desktop computers, laptop computers, and mobile devices, such as cell phones and PDAs, that enable you to organize the daily stream of information that comes across your desk (e.g., appointments, weekly meeting reminders, to-do lists) in a manner that suits your personal style and to retrieve it when needed. Table 6.4 provides a small sampling of PIMSs and some of their advertising claims. A PIMS is a good idea. Unfortunately, most systems are not well implemented, do not live up to their claims, and have not been subjected to serious user testing. Following are a number of guidelines that should help in designing and using such systems:

- **Plan for retrieval.** There is no use in filing things if you cannot retrieve them or if you do not have a need for them in the future. All storage should be retrieval minded. Will I need it, and if so, under what circumstances? If I do need it, how will I find it?
- **Organization is essential.** Long-term memory works well because its contents are self-organized by categories, associations, features, etc. The same is true for external memory. Organization may be generated at input (preprocessing) by filing and tagging the information while entering it or later (postprocessing) when one has time for housekeeping. The worst time for organization is at retrieval. The problem with organization is that it takes
Table 6.4. Text Advertising for Several Brands of PIMSs

<table>
<thead>
<tr>
<th>Brand</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>TreePad</td>
<td><a href="http://www.treepad.com/">www.treepad.com/</a></td>
</tr>
<tr>
<td>Task Plus</td>
<td><a href="http://www.contactplus.com/products/freestuff/task.htm">www.contactplus.com/products/freestuff/task.htm</a></td>
</tr>
<tr>
<td>Personal Knowbase</td>
<td><a href="http://www.bitsmithsoft.com/">www.bitsmithsoft.com/</a></td>
</tr>
<tr>
<td>PIM eCentral</td>
<td><a href="http://www.softspecialist.com/Browser-and-PIM-eCentral-1111/Browser-and-PIM-eCentral.htm">www.softspecialist.com/Browser-and-PIM-eCentral-1111/Browser-and-PIM-eCentral.htm</a></td>
</tr>
</tbody>
</table>

TreePad saves you time, allowing you to keep your notes, documents, hyperlinks and images at the brief distance of a click. Download TreePad and enjoy better access, organization, and control over your data, notes, bills, projects, clients, addresses, letters, speeches, research, collections, classroom notes, Web pages, links, bibliographic listings, and whatever else your creativity enables you to entrust TreePad with, for good organization, easy access, and safe storage.

Task Plus is the best task and calendar management program designed for simplicity and speed. Running in your system tray, Task Plus keeps track of an unlimited number of personal to-do items as well as date rated appointments plus holidays and repeating events such as birthdays. Using sophisticated alarms, it can also play sound (WAV file) when a task comes due! Powerful filters make it easy to view tasks in certain categories or by project.

Personal Knowbase makes your life easier. Take control of your notes. Personal Knowbase is the note organizer software that makes it easy to manage the large amount of information that crosses your desk every day. This free-form notes manager uses a natural way to retrieve notes, using keywords to filter your knowledge base for related information quickly. Whether you are organizing research notes, e-mail archives, or story ideas, Personal Knowbase makes your life easier.

PIM eCentral is the most user-friendly, most fun to use all-in-one desktop productivity tool. With eCentral, you can quickly and easily organize your time and contacts, and browse the Web with no pop-ups. At the same time, you can listen to your favorite MP3 songs.

Tools included: Calendar and organizer with reminders and to-do list, address book, Web browser with pop-up blocker, built in Windows Desktop and Explorer access, e-mail program launcher, Windows Calculator launcher, calendar photo personalization, music player (MP3), video player (AVI, MPEG). The personal information manager can be used as a combined calendar, diary, organizer, and reminder.

时间。自我组织的系统是需要的，与之相关的组织工具和代理来处理你的工作。

- **Multiple systems are a problem.** If personal information is stored on different devices and systems that do not talk to each other, the difficulty of finding the information is multiplied. You may have to check a number of different systems in different places. Most PDAs have programs to sync the information, but they are not perfect and not always convenient.

- **Backup information.** However, if information is only on one device, it can be lost. All PIMSs must have multiple backups that are easy to use and convenient, if not automatic. Restoring information from the backup should be reliable and automatic.

- **Set security levels.** As you retrieve information from your long-term memory, you are keenly aware of whether it is private or public. Some PIMSs lock all information and require a password to access it. So, even to find out whether May 19, 1979, was a Saturday, you have to enter your password.
Others allow you to lock individual bits of information, but this can be quite time consuming.

- **Use multiple and appropriate media.** Today’s PIMSs and PDAs allow multiple media (e.g., text, voice, pictures, video). Sometimes voice input of a name and phone number is superior to text input or a picture of a business card is superior to voice or text. The problem will be resolving the media during postprocessing of the information.

- **Rugged, reliable, and always present.** We joke, “If your head wasn’t attached to your body, you’d probably forget it at home.” A PIMS is only good if you have it with you – in the office, at home, while hiking, or even while swimming. It must be durable, easily locatable, and wearable.

- **Usability is key.** If something is too hard to figure out, it will not be used. PIMSs and PDAs must make it easy to store, maintain, and retrieve information. This includes use of input devices such as the keyboard or touchscreen, readability of the screen, and all menu navigation. The more that is automatic (e.g., autoentry of time and date, autoupdating, autotransmission of V-cards) and intuitive, the better.

### Using What Computers Remember

**Human memory is purpose driven.** We remember things for many different reasons – aesthetic memories that we enjoy, practical information that will definitely prove to be useful, and other information that might be useful. When our memory fails, we often turn to others’ memories or to artifacts left from our activities. As it turns out, computers record a lot of information about the user. The computer stores many of our activities at the interface. In fact, it is rather shocking how much episodic memory can be retrieved from e-mail archives, phone records, credit card transactions, online shopping, blogs, server logs, history files, etc. If we are having a hard time trying to remember something in the past, we may be able to troll through the computer’s memory to find it. Increasingly, software tools are being developed to help us do just that.

Some psychologists believe that our brains store memories of all stimuli and events that occur, but that we are just not able to recall all of them. The same may be true of computer storage. Even if events are stored in files to which we have access, it may be difficult to find the one we want. In addition, there are many files to which we do not have access, such as those held by Internet providers and financial and government organizations. Nevertheless, with the right retrieval tools, the information may be found.

### End Thoughts

Learning and memory have been of great concern to psychologists. A number of theories have been developed, and thousands of studies have been
conducted. Our world requires us to learn, memorize, and recall a tremendous amount of information. Computers and the human–computer interface add substantially to this information. The challenge is to design interfaces that minimize the sheer volume of information that people have to learn and recall, and make it easy to learn and recall what information they do need.

Computers also provide solutions to our problems with memory. When human memory fails, computer memory may kick in to help us out. External memory is a type of assistive technology, which we talk more about in Chapter 14. Learning and memory are highly purpose driven in pursuit of solving problems. We remember things to accomplish tasks, to satisfy needs, and for our own enjoyment. Chapter 7 explores the issue of problem solving.

Suggested Exercises

1. Consider a simple computer program like Notepad. What did you have to learn to use this program?

2. If you have upgraded or are upgrading a program or an OS, see if you can count the number of deleted, added, or changed options, commands, icons, etc., in the two versions. How much positive, neutral, and negative transfer do you believe there will be?

3. Consider two different systems or versions of an application, and identify the types of transfer going from one to the other according to Table 6.3.

References


