2.1. THE MODEL HUMAN PROCESSOR

A computer engineer describing an information-processing system at the systems level (as opposed, for instance, to the component level) would talk in terms of memories and processors, their parameters and interconnections.[1] By suppressing detail, such a description would help him to envision the system as a whole and to make approximate predictions of gross system behavior.

The human mind is also an information-processing system, and a description in the same spirit can be given for it. The description is approximate when applied to the human, intended to help us remember facts and predict user-computer interaction rather than intended as a statement of what is really in the head. But such a description is useful for making approximate predictions of gross human behavior. We therefore organize our description of the psychological science base around a model of this sort. To distinguish the simplified account of the present model from the fuller psychological theory we would present in other contexts, we call this model the *Model Human Processor.*

The Model Human Processor (see Figures 2.1 and 2.2) can be described by (1) a set of memories and processors together with (2) a set of principles, hereafter called the "principles of operation." Of the two parts, it is easiest to describe the memories and processors first, leaving the description of the principles of operation to arise in context.

The Model Human Processor can be divided into three interacting subsystems: (1) the *perceptual system,* (2) the *motor system,* and (3) the *cognitive system,* each with its own memories and processors. The perceptual system consists of sensors and associated buffer memories, the most important buffer memories being a Visual Image Store and an Auditory Image Store to hold the output of the sensory system while it is being symbolically coded. The cognitive system receives symbolically coded information from the sensory image stores in its Working Memory and uses previously stored information in Long-Term Memory to make decisions about how to respond. The motor system carries out the response. As an approximation, the information processing of the human will be described as if there were a separate processor for each subsystem: a Perceptual Processor, a Cognitive Processor, and a Motor Processor. For some tasks (pressing a key in response to a light) the human must behave as a serial processor. For other tasks (typing, reading, simultaneous translation) integrated, parallel operation of the three subsystems is possible, in the manner of three pipelined processors: information flows continuously from input to output with a characteristically short time lag showing that all three processors are working simultaneously.

[1] For a survey of computing systems in these terms see Siewiorek, Bell, and Newell (1981).

The memories and processors are described by a few parameters. The most important parameters of a memory are

μ, the storage capacity in items,
δ, the decay time of an item, and
κ, the main code type (physical, acoustic, visual, semantic).

The most important parameter of a processor is

τ, the cycle time.

Whereas computer memories are usually also characterized by their access time, there is no separate parameter for access time in this model since it is included in the processor cycle time.

We now consider each of the subsystems in more detail.

The Perceptual System

The perceptual system carries sensations of the physical world detected by the body's sensory systems into internal representations of the mind by means of integrated sensory systems. An excellent example of the integration of a sensory system is provided by the visual system: The retina is sensitive to light and records its intensity, wave length, and spatial distribution. Although the eye takes in the visual scene over a wide angle, not quite a full half-hemisphere, detail is obtained only over a narrow region (about 2 degrees across), called the *fovea.* The remainder of the retina provides peripheral vision for orientation. The eye is in continual movement in a sequence of saccades, each taking about 30 msec to jump to the new point of regard[2] and dwelling there 60~700 msec for a total duration of

[2] Russo (1978).

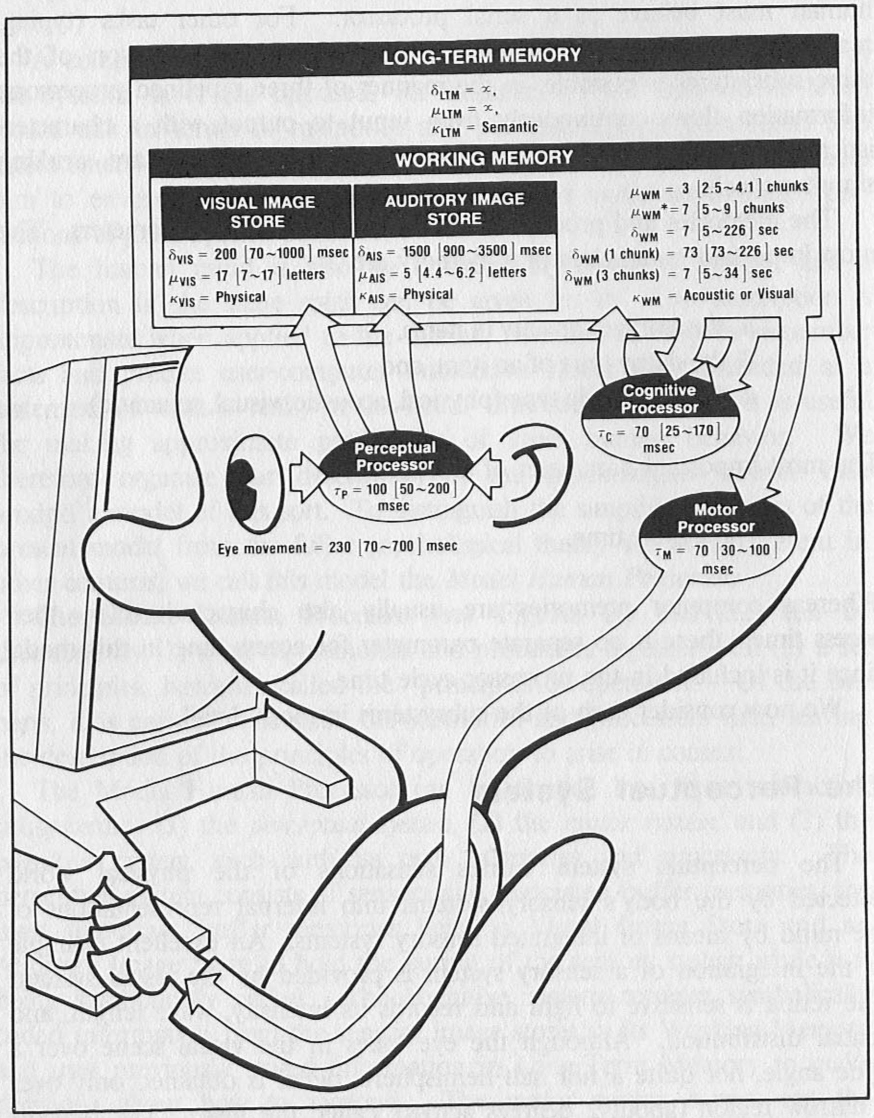

Figure 2.1. The Model Human Processor—memories and processors. Sensory information flows into Working Memory through the Perceptual Processor. Working Memory consists of activated chunks in Long-Term Memory. The basic principle of operation of the Model Human Processor is the *Recognize-Act Cycle of the Cognitive Processor* (P0 in Figure 2.2). The Motor Processor is set in motion through activation of chunks in Working Memory.

P0. ***Recognize-Act Cycle of the Cognitive Processor.*** On each cycle of the Cognitive Processor, the contents of Working Memory initiate actions associatively linked to them in Long-Term Memory; these actions in turn modify the contents of Working Memory.

P1. ***Variable Perceptual Processor Rate Principle.*** The Perceptual Processor cycle time τ_P varies inversely with stimulus intensity.

P2. ***Encoding Specificity Principle.*** Specific encoding operations performed on what is perceived determine what is stored, and what is stored determines what retrieval cues are effective in providing access to what is stored.

P3. ***Discrimination Principle.*** The difficulty of memory retrieval is determined by the candidates that exist in the memory, relative to the retrieval clues.

P4. ***Variable Cognitive Processor Rate Principle.*** The Cognitive Processor cycle time τ_c is shorter when greater effort is induced by increased task demands or information loads; it also diminishes with practice.

P5. ***Fitts's Law.*** The time T_{pos} to move the hand to a target of size S which lies a distance D away is given by:

$$T_{pos} = I_M \log_2 (D/S + .5)\,, \tag{2.3}$$

where I_M = 100 [70~120] msec/bit.

P6. ***Power Law of Practice.*** The time T_n to perform a task on the nth trial follows a power law:

$$T_n = T_1 n^{-\alpha}\,, \tag{2.4}$$

where α = .4 [.2~.6].

P7. ***Uncertainty Principle.*** Decision time T increases with uncertainty about the judgement or decision to be made:

$$T = I_C H\,,$$

where H is the information-theoretic entropy of the decision and I_C = 150 [0~157] msec/bit. For n equally probable alternatives (called Hick's Law),

$$H = \log_2 (n + 1)\,. \tag{2.8}$$

For n alternatives with different probabilities, p_i, of occurence,

$$H = \Sigma_i p_i \log_2 (1/p_i + 1)\,. \tag{2.9}$$

P8. ***Rationality Principle.*** A person acts so as to attain his goals through rational action, given the structure of the task and his inputs of information and bounded by limitations on his knowledge and processing ability:

Goals + Task + Operators + Inputs
+ Knowledge + Process-limits → Behavior

P9. ***Problem Space Principle.*** The rational activity in which people engage to solve a problem can be described in terms of (1) a set of states of knowledge, (2) operators for changing one state into another, (3) constraints on applying operators, and (4) control knowledge for deciding which operator to apply next.

Figure 2.2. The Model Human Processor—principles of operation.

$$\text{Eye-movement} = 230\,[70\sim700]\ \text{msec}\,.^{3}$$

(In this expression, the number 230 msec represents a typical value and the numbers in brackets indicate that values may range from 70 msec to 700 msec depending on conditions of measurement, task variables, or subject variables.) Whenever the target is more than about 30 degrees away from the fovea, head movements occur to reduce the angular distance. These four parts—central vision, peripheral vision, eye movements, and head movements—operate as an integrated system, largely automatically, to provide a continual representation of the visual scene of interest to the perceiver.

PERCEPTUAL MEMORIES

Very shortly after the onset of a visual stimulus, a representation of the stimulus appears in the *Visual Image Store* of the Model Human Processor. For an auditory stimulus, there is a corresponding *Auditory Image Store.* These sensory memories hold information coded *physically,* that is, as an unidentified, non-symbolic analogue to the external stimulus. This code is affected by physical properties of the stimulus, such as intensity. For our purposes we need not enter into the details of the physical codes for the two stores but can instead just write:

$$\kappa_{VIS} = \text{physical}\,,$$
$$\kappa_{AIS} = \text{physical}\,.$$

For example, the Visual Image Store representation of the number 2 contains features of curvature and length (or equivalent spatial frequency patterns) as opposed to the recognized digit.

The perceptual memories are intimately related to the cognitive Working Memory as Figure 2.1 depicts schematically. Shortly after a physical representation of a stimulus appears in one of the perceptual memories, a recognized, symbolic, acoustically-coded (or visually-coded)

[3] Actual saccadic eye-movement times (travel + fixation time) can vary quite considerably depending on the task and the skill of the observer. Russo (1978, Table 2, p. 94) lists 70 msec as the minimum time and 230 msec as a typical time. The largest time given by Busswell (1922, p. 31) for eye-movements in reading is 660 msec (for first-grade children), which we round to 700 msec.

representation of at least part of the perceptual memory contents occurs in Working Memory. If the contents of perceptual memory are complex or numerous (for example, an array of letters) and if the stimulus is presented only fleetingly, the perceptual memory trace fades, and Working Memory is filled to capacity before all the items in the perceptual memory can be transferred to representations in Working Memory (for letters the coding goes at about 10 msec/letter). However, the Cognitive Processor can specify which portion of the perceptual memory is to be so encoded. This specification can only be by physical dimensions, since this is the only information encoded: after being shown a colored list of numbers and letters, a person can select (without first identifying what number or letter it is) the top half of the Visual Image Store or the green items, but not the even digits or the digits rather than the letters.

Figure 2.3 shows the decay of the Visual Image Store and the Auditory Image Store over time. As an index of decay time, we use the half-life, defined as the time after which the probability of retrieval is less than 50%. While exponential decay is not necessarily implied by the use of the half-life, Figure 2.3 shows that it is often a good approximation to the observed curves. The Visual Image Store has a half-life of about

$$\delta_{VIS} = 200\,[90\sim1000]\ \text{msec}\,,^{4}$$

but the Auditory Image Store decays more slowly,

[4] A least-squares fit to data estimated from figures appearing in Sperling (1960) and Averbach and Coriell (1961) yields the following facts. The half-life of the letters in excess of the memory span that subjects could report in the partial report condition of Sperling's (1960) experiment was 621 msec (9-letter stimulus) and 215 msec (12-letter stimulus). Averbach and Coriell's (1961) experiment gives a half-life of 92 msec (16-letter stimulus). The typical value for δ_{VIS} has been set at 200 msec, representing the middle of these. The lower and upper bounds for δ_{VIS} are set at rounded-off values reflecting the fastest subject in the condition with the shortest half-life and the slowest subject in the condition with the longest half-life. The shortest half-life in these experiments was 93 msec for Averbach and Coriell's Subject GM (16-letter condition); the longest half-life was 940 msec for Sperling's Subject ROR (9-letter condition). It is possible to have the average half-life be 92 msec, shorter than the half-life of any subject, because this average is computed by first taking the mean of each point across subjects, then computing the slope of the best least-square fitting line in semilog coordinates.

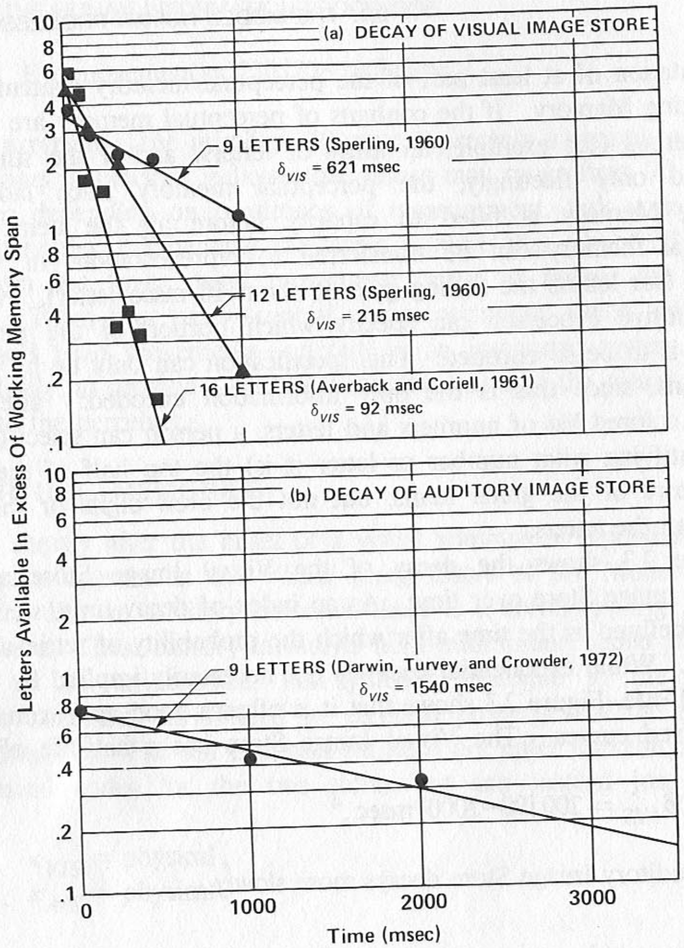

Figure 2.3. Time decay of Visual and Auditory Image Stores. (a) Decay of the Visual Image Store. In each experiment, a matrix of letters was made observable tachistoscopically for 50 msec. In the case of the Sperling experiments, a tone sounded after the offset of the letters to indicate which row should be recalled. In the case of the Averbach and Coriell experiment, a bar appeared after the offset of the letters next to the letter to be identified. The percentage of indicated letters that could be recalled eventually asymptotes to μ_{WM}*. The graph plots the percentage of letters reported correctly in excess of μ_{WM}* as a function of time before the indicator.
(b) Decay of the Auditory Image Store. Nine letters were played to the observers over stereo earphones arranged so that three sequences of letters appear to come from each of three directions. A light lit after the offset of the letters to indicate which sequence should be recalled. The graph plots the percentage of the relevant 3-letter sequence in excess of μ_{WM}* reported correctly as a function of time before the light was lit.

$$\delta_{AIS} = 1500\,[900\sim3500]\ \text{msec}\,,^{5}$$

consistent with the fact that auditory information must be interpreted over time. The capacity of the Visual Image Store is hard to fix precisely but for rough working purposes may be taken to be about

$$\mu_{VIS} = 17\,[7\sim17]\ \text{letters}\,.^{6}$$

The capacity of the Auditory Image Store is even more difficult to fix, but would seem to be around

$$\mu_{AIS} = 5\,[4.4\sim6.2]\ \text{letters}\,.^{7}$$

PERCEPTUAL PROCESSOR

The cycle time τ_P of the Perceptual Processor is identifiable with the so-called *unit impulse response* (the time response of the visual system to

[5] The half-life of the letters in excess of the memory span that subjects could report in the partial report condition of Darwin, Turvey, and Crowder's (1972) experiment was 1540 msec, which we have rounded to δ_{AIS} = 1500 msec. The difference in decay half-life as a function of letter order in their experiment (963 msec for the third letter, 3466 msec for the first letter) has been rounded to give lower and upper bounds of 900 and 3500. Other techniques have been used to obtain values for the "decay time" of the Auditory Image Store. For example, use of a masking technique gives estimates of around 250 msec full decay (Massaro, 1970), but these experiments have been criticized by Klatzky (1980, p. 42) because they may only measure the time necessary to transmit categorical information to Working Memory. On the other end, experiments that measure the delay at which there is still some facilitation of the identification of a noisy signal (Crossman, 1958; Guttman and Julesz, 1963) give very wide full-decay estimates: from 1000 msec to 15 minutes!

[6] Sperling (1963, p. 22) estimates the capacity of the Visual Image store in terms of the number of letters available at least 17 letters and possibly more. The fewest number of letters available for any subject immediately after stimulus presentation in the 9-letter condition (Sperling, 1960) was 7.4 letters for Subject NJ.

[7] Range is from the number of letters or numbers that could be reported by Darwin, Turvey, and Crowder's (1972) subjects in an experiment in which they had to give the trio of letters coming from one of three directions (indicated by a visual cue shortly after the end of the sounds). Lowest value, 4.4 letters, is for accuracy of recalling second letter of triple when subjects had to name all items coming from a certain direction (Figure 1, p. 259). Highest number, 6.2 letters, is for recall by category when no location was required (Figure 2(B), p. 262).

a very brief pulse of light)[8] and its duration is on the order of

$$\tau_P = 100\,[50\sim200]\ \text{msec}.^9$$

If a stimulus impinges upon the retina at time $t = 0$, at the end of time $t = \tau_P$ the image is available in the Visual Image Store and the human claims to see it. In truth, this is an approximation, since different information in the image becomes available at different times, much as a photograph develops.[10] For example, movement information and low spatial frequency information are available sooner than other information. A person can react before the image is fully developed or can wait for a better image, according to whether speed or accuracy is the more important.

Perceptual events occurring within a single cycle are combined into a single percept if they are sufficiently similar. For example, two lights occurring at different nearby locations within 60~100 msec combine to give the impression of a single light in motion. A brief pulse of light, lasting t msec with intensity I, has the same appearance as a longer pulse of less-intense light, provided both pulses last less than 100 msec, giving rise to Bloch's Law (1885):

$$I \cdot t = k, \quad t < \tau_P.$$

Two brief pulses of light within a cycle combine their intensities in a more complicated way, but still give a single percept.[11] Thus there is a basic quantum of experience; and the present is not an instantaneous dividing line between past and future, but has itself duration.

Figure 2.4 shows the results of an experiment in which subjects were presented with a rapid set of clicks, from 10 to 30 clicks per second, and were asked to report how many they heard. The results show that they heard the correct number when the clicks were presented at 10 clicks/sec, but missed progressively more clicks at 15 and 30 clicks/sec. A simple

8 See Ganz (1975).

9 The source of the range is the review by Harter (1967), who also discusses the suggestion that the cycle time can be identified with the 77~125 msec alpha period in the brain.

10 See Ericksen and Shultz (1978), Ganz (1975).

11 See Ganz (1975).

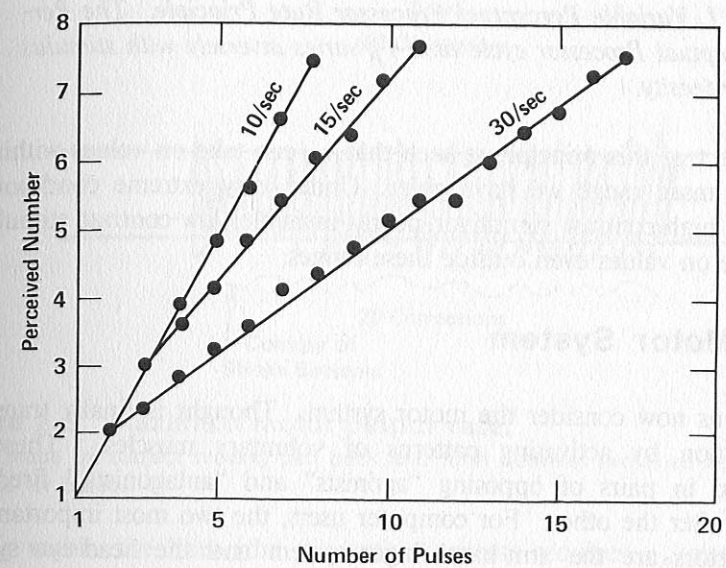

Figure 2.4. Fusion of clicks within 100 msec. A burst of sound containing an unknown number of auditory clicks at the uniform rate of 10/sec, 15/sec, or 30/sec was presented to the subject. The graph plots the number of clicks/burst reported as a function of the number presented. After Cheatham and White (1954, Figure 1, p. 427).

analysis in terms of the Model Human Processor shows why. When the experimenter plays the clicks at 10 clicks/sec, there is one click for each $\tau_P \simeq 100$ msec interval and the subject hears each click. But when the experimenter plays the clicks at 30 clicks/sec, the three clicks in each 100 msec cycle time are fused into a single percept (perhaps sounding a little louder) and the subject hears only one click instead of three, or 10 clicks/sec. The data in Figure 2.4 show that the number of clicks/sec perceived by the subjects does in fact stay approximately constant in the 10 clicks/sec range (the measured values of the slopes are 9~11 clicks/sec) for the three rates of presentation.

As a second-order phenomenon, the processor time τ_P is not completely constant, but varies somewhat according to conditions. In particular, τ_P is shorter for more intense stimuli, a fact derivable from a more detailed examination of the human information-processor using linear systems theory, but which we simply adopt as one of the principles of operation (Figure 2.2):

P1. Variable Perceptual Processor Rate Principle. *The Perceptual Processor cycle time τ_P varies inversely with stimulus intensity.*

The effect of this principle is such that τ_P can take on values within the 50~200 msec range we have given. Under very extreme conditions of intense, high-contrast stimuli or nearly invisible, low-contrast stimuli, τ_P can take on values even outside these ranges.

The Motor System

Let us now consider the motor system. Thought is finally translated into action by activating patterns of voluntary muscles. These are arranged in pairs of opposing "agonists" and "antagonists," fired one shortly after the other. For computer users, the two most important sets of effectors are the arm-hand-finger system and the head-eye system.

Movement is not continuous, but consists of a series of discrete micromovements, each requiring about

$$\tau_M = 70\,[30\sim100]\text{ msec},$$ [12]

which we identify as the cycle time of the Motor Processor. The feedback loop from action to perception is sufficiently long (200~500 msec) that rapid behavioral acts such as typing and speaking must be executed in bursts of preprogrammed motor instructions.

An instructive experiment is to have someone move a pen back and forth between two lines as quickly as possible for 5 sec (see Figure 2.5). Two paths through the processors in Figure 2.1 are clearly visible: (1) The Motor Processor can issue commands ("open loop") about once every τ_M = 70 msec; in Figure 2.5 this path leads to the 68 pen reversals made by the subject in the 5 sec interval, or τ_M = 74 msec/reversal. (2) The subject's perceptual system can perceive whether the strokes are

12 The limit of repetitive movement of the hand, foot, or tongue is about 10 movements/sec (Fitts and Posner, 1967, p. 18). Chapanis, Garner, and Morgan (1949, p. 284) cite tapping rates of 8~13 taps/sec (38~62 movements/sec, assuming 2 movements/tap). Fox and Stansfield (1964) cite figures of 130 msec/tap = 65 msec/movement. Repetition of the same key in Kinkead's data (Figure 2.15*b*) averages to 180 msec/keystroke = 90 msec/movement. The scribbling rate in Figure 2.5 was 74 msec/movement. We summarize these as 70 [30~100] msec/movement.

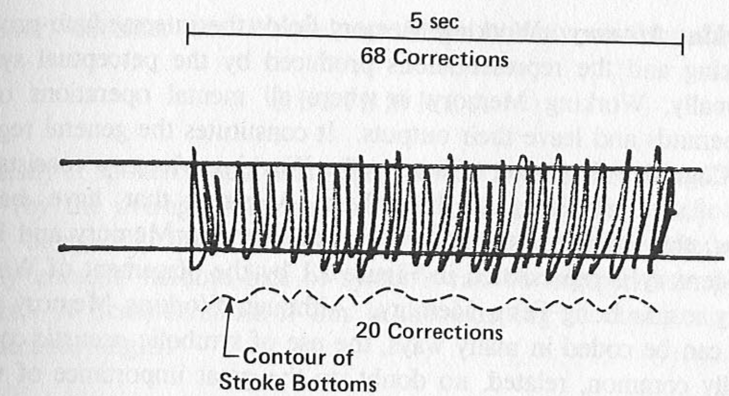

Figure 2.5. Maximum motor output rate.
Marks made by subject moving pen back and forth between two lines as fast as possible for 5 sec.

staying within the lines (the perception process requires τ_P msec) and send this information to the cognitive system, which can then advise (the decision process requires τ_C msec) the motor system to issue a correction (the motor process requires τ_M msec). The total time, therefore, to make a correction using visual feedback ("closed loop") should be on the order of $\tau_P + \tau_C + \tau_M$ = 240 msec; in Figure 2.5, this path leads to the roughly 20 corrections about the ruled guidelines as indicated by the dotted line tracing the contours of the bottoms of the strokes, or (5 sec)/(20 movements) = 250 msec/movement.

The Cognitive System

In the simplest tasks, the cognitive system merely serves to connect inputs from the perceptual system to the right outputs of the motor system. But most tasks performed by a person are complex and involve learning, retrieval of facts, or the solution of problems. As would be expected, the memories and the processor for the cognitive system are more complicated than those for the other systems.

COGNITIVE MEMORIES

There are two important memories in the cognitive system: a *Working Memory* to hold the information under current consideration and a *Long-Term Memory* to store knowledge for future use.

Working Memory. Working Memory holds the intermediate products of thinking and the representations produced by the perceptual system. Functionally, Working Memory is where all mental operations obtain their operands and leave their outputs. It constitutes the general registers of the Cognitive Processor. Structurally, Working Memory consists of a subset of the elements in Long-Term Memory that have become *activated;* this intimate association between Working Memory and Long-Term Memory is represented in Figure 2.1 by the placement of Working Memory inside Long-Term Memory. Although Working Memory information can be coded in many ways, the use of symbolic *acoustic* codes is especially common, related, no doubt, to the great importance of verbal materials to the tasks people frequently perform. The user of a telephone, for example, is especially liable to dial numbers mistakenly that sound like the numbers he has just looked up. *Visual* codes, if required by the the task, are also possible (as are some other types of codes). For purposes of the Model Human Processor we consider the predominant code types to be

$$\kappa_{WM} = \text{acoustic or visual}.$$

It is important to distinguish the symbolic, nonphysical acoustic or visual codes of Working Memory, which are unaffected by physical parameters of the stimulus (such as intensity), from the nonsymbolic, physical codes of the sensory image stores, which *are* affected by physical parameters of the stimulus.

The activated elements of Long-Term Memory, which define Working Memory, consist of symbols, called *chunks,* which may themselves be organized into larger units. It is convenient to think of these as nested abstract expressions: CHUNK1 = (CHUNK2 CHUNK3 CHUNK4), with, for instance, CHUNK4 = (CHUNK5 CHUNK6).[13] What constitutes a chunk is as much a function of the user as of the task, for it depends on the contents of the user's Long-Term Memory. The sequence of nine letters below is beyond the ability of most people to repeat back:

B C S B M I C R A

13 It is also possible to think of these as semantic networks, such as those in Anderson (1980) and other recent publications. At the level of our discussion, any of these notations will suffice about equally well. See also Simon (1974) for a technical definition of chunk.

However, consider the list below, which is only slightly different:

C B S I B M R C A

Especially if spoken aloud, this sequence will be chunked into CBS IBM RCA (by the average American college sophomore) and easily remembered, being only three chunks. If the user can perform the recoding rapidly enough, random lists of symbols can be mapped into prepared chunks. A demonstration of this is the mapping of binary digits into hexadecimal digits:

0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 1 0 1 1 0 0 1 1 0 1 0 0 0
0100 0010 0001 0011 0110 0110 1000
4 2 1 3 6 6 8

This last can be easily remembered. The coding must be done in both directions, binary to hexadecimal and hexadecimal to binary, and takes substantial practice before it can be carried out as part of a regular memory-span test, but it can be done. Indeed, with extended effort, the digit span can be increased enormously. A Carnegie-Mellon University student holds the current record at 81 decimal digits, presented at a uniform rate of 1 digit per second.[14] This particular event occurred as part of a psychological study, where it could be verified that all the gain was due to elaborate recoding and immense practice in its use and development, rather than any physiological endowment.

Chunks can be related to other chunks. The chunk ROBIN, for example, sounds like the chunk ROBERT. It is a subset of the chunk BIRD, it has chunk WINGS, it can chunk FLY. When a chunk in Long-Term Memory is activated, the activation spreads to related chunks and to chunks related to those. As the activation spreads to new chunks, the previously activated chunks become less accessible, because there is a limited amount of activation resource. The new chunks are said to *interfere* with the old ones. The effect of this interference is that the chunk appears to fade from Working Memory with time (unless reactivated), as the decay curves in Figure 2.6 show. The curves are significantly affected by other variables, including the number of other chunks the user is trying to remember, retrieval interference with similar

14 Ericsson, Chase, and Faloon (1980); Chase and Ericsson (1981).

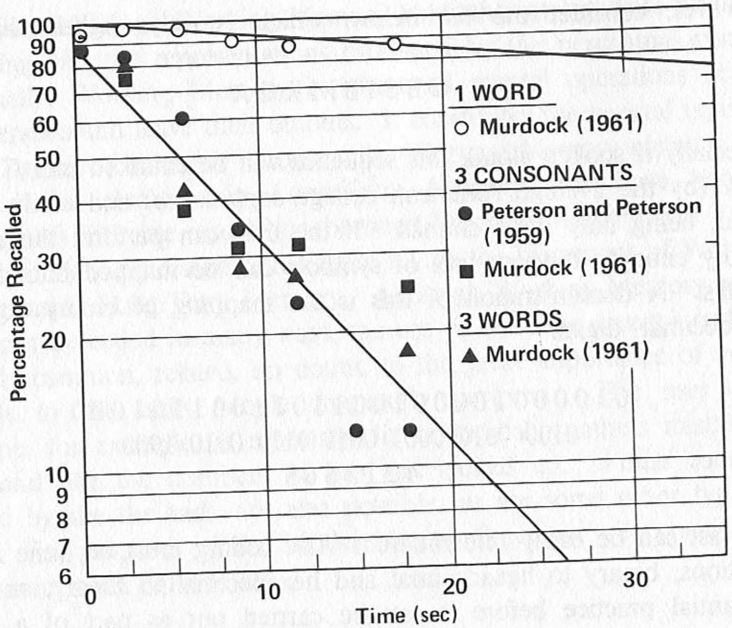

Figure 2.6. Working Memory decay rate. He counts Subject is given either one or three words or consonants to remember. backwards (preventing rehearsal) for a time and then recalls stimulus. Graph plots proportion of items correctly recalled as a function of the time elapsed until recall began.

chunks in Working Memory, and input and retrieval memory strategies of the user. As a working value we take the half-life of 7 sec from the curve in Figure 2.6, which together with other data gives

$$\delta_{WM} = 7\ [5\sim226]\ \text{sec}.$$ [15]

The decay parameter δ_{WM} has a wide range, because most of the apparent decay comes about from the details of interference, as we have noted above. But these details are difficult to analyze, so it is most convenient to accept the range and talk in terms of decay. Since the

15 For three chunks, Peterson and Peterson's (1959) data (Figure 2.6) give a half-life of about 5 sec. Murdock's data (Murdock, 1961) in Figure 2.6 give a half-life of about 7 sec for 3 words and also 9 sec for 3 consonants. On the other hand, Melton's (1963) data give a much longer half-life of 34 sec. For one chunk, Murdock's data in Figure 2.6 and Melton's (1963) give half-lives of 73 sec and 226 sec, respectively.

decay rate is particularly sensitive to the number of chunks in the recalled item, it is useful to record the decay rate of representative item sizes:

$$\delta_{WM}(\text{1 chunk}) = 73\ [73\sim226]\ \text{sec},$$ [15]
$$\delta_{WM}(\text{3 chunks}) = 7\ [5\sim34]\ \text{sec}.$$ [15]

When people are asked to recall information a few seconds after hearing it, they use both Working Memory and Long-Term Memory to do so. Experimentally, these two systems have been teased apart showing that there is a *pure capacity of Working Memory* (example: number of immediately preceding digits recallable from a long series when the series unexpectedly stops),

$$\mu_{WM} = 3\ [2.5\sim4.1]\ \text{chunks}.$$ [16]

When this pure capacity is augmented by the use of Long-Term Memory, the *effective capacity of Working Memory* $\mu_{WM}{}^*$ (example: longest number that can be repeated back) extends to the familiar 7±2 chunks,

$$\mu_{WM}{}^* = 7\ [5\sim9]\ \text{chunks}.$$ [17]

Long-Term Memory. Long-Term Memory holds the user's mass of available knowledge. It consists of a network of related chunks, accessed associatively from the contents of the Working Memory. Its contents comprise not only facts, but procedures and history as well.

Apparently, there is no erasure from Long-Term Memory,

$$\delta_{LTM} = \infty.$$

However, successful retrieval of a chunk depends on whether associations to it can be found. There are two reasons the attempt to retrieve a chunk might fail: (1) effective retrieval associations cannot be found, or

16 Crowder (1976) reviews several methods. Estimates are Waugh and Norman (1965) method, 2.5 items; Raymond (1969) method, 2.5 items; Murdock (1960*b*, 1967) method, 3.2~4.1 items; Tulving and Colatla (1970) method, 3.3~3.6 items. See also Glanzer and Razel (1974).

17 Miller (1956).

(2) similar associations to several chunks interfere with the retrieval of the target chunk. The great importance of these links between particular chunks in Long-Term Memory, that is, the *semantic* coding of information, leads us to list it as the predominant code type,

$$\kappa_{LTM} = \text{semantic}.$$

To be stored in Long-Term Memory, information from the sensory memories must ultimately be encoded into symbolic form: a pattern of light and dark might be coded as the letter A, an extended pattern coded as a system error message. When the information from Working Memory becomes part of Long-Term Memory, the precise way in which it and the coincident Working Memory contents were encoded determines what cues will be effective in retrieving the item later. Suppose a user names a computer-imaging file **LIGHT** (as opposed to **DARK**). If he later scans a directory listing of file names to identify which ones were the ones he created and thinks of **LIGHT** (as opposed to **HEAVY**), he will not be able to recognize the file, because he will be using a different set of retrieval cues. As a principle of operation,

> ***P2. Encoding Specificity Principle.***[18] *Specific encoding operations performed on what is perceived determine what is stored, and what is stored determines what retrieval cues are effective in providing access to what is stored.*

Because of interference with other chunks in memory that are more strongly activated by the associations used as retrieval cues, information, despite being physically present, can become functionally lost. Stated as a principle,

> ***P3. Discrimination Principle.*** *The difficulty of memory retrieval is determined by the candidates that exist in the memory, relative to the retrieval cues.*

Items cannot be added to Long-Term Memory directly (accordingly, Figure 2.1 shows no arrow in this direction); rather, items in Working

18 Tulving and Thompson (1973).

Memory (possibly consisting of several chunks) have a certain probability of being retrievable later from Long-Term Memory. The more associations the item has, the greater its probability of being retrieved. If a user wants to remember something later, his best strategy is to attempt to associate it with items already in Long-Term Memory, especially in novel ways so there is unlikely to be interference with other items. Of course this activity, by definition, activates more items in Long-Term Memory, causing new items to appear in Working Memory, and use capacity. On a paced task, where a user is given items to remember at a constant rate, the percentage of the items recalled later increases as the time/item increases (the probability the item will be stored in Long-Term Memory and linked so it can be retrieved increases with residence time in Working Memory), until the time allowed per item is of the same magnitude as the decay time of Working Memory (after which, more time available for study does not increase the time the item is in Working Memory), around δ_{WM} sec/chunk = 7 sec/chunk.[19]

Storing new chunks in Long-Term Memory thus requires a fair amount of time and several Long-Term Memory retrievals. On the other hand, Long-Term Memory is accessed on every 70 msec cognitive-processing cycle. Thus the system operates as a fast-read, slow-write system. This asymmetry puts great importance on the limited capacity of Working Memory, since it is not possible in tasks of short duration to transfer very much knowledge to Long-Term Memory as a working convenience.

COGNITIVE PROCESSOR

The *recognize-act cycle*, analogous to the fetch-execute cycle of standard computers, is the basic quantum of cognitive processing. On each cycle, the contents of Working Memory initiate associatively-linked actions in Long-Term Memory ("recognize"), which in turn modify the contents of Working Memory ("act"), setting the stage for the next cycle. Plans, procedures, and other forms of extended organized behavior are built up out of an organized set of recognize-act cycles.

Like the other processors, the Cognitive Processor seems to have a cycle time of around a tenth of a second:

19 Newell and Simon (1972, p. 793) reviews experiments that gives times of 8~13 sec/chunk.

$$\tau_C = 70\,[25\sim170]\ \text{msec}.$$ [20]

The cycle times for several types of tasks are given in Figure 2.7. The times vary in the 25~170 msec/cycle range, depending on the specific experimental phenomenon and experimental circumstances with which one wishes to identify the cycle. We have chosen as a nominal value 70 msec, about at the median of those in Figure 2.7, but have included within the upper and lower limits all the estimates from the figure. As with the Perceptual Processor, the cycle time is not constant, but can be shortened by practice, task pacing, greater effort, or reduced accuracy.

P4. Variable Cognitive Processor Rate Principle. *The Cognitive Processor cycle time τ_C is shorter when greater effort is induced by increased task demands or information loads; it also diminishes with practice.*

The cognitive system is fundamentally parallel in its recognizing phase and fundamentally serial in its action phase. Thus the cognitive system can be aware of many things, but cannot do more than one deliberate thing at a time. This seriality occurs on top of the parallel activities of the perceptual and motor systems. Driving a car, reading roadside advertisements, and talking can all be kept going by skilled intermittent allocation of control actions to each task, along the lines of familiar interrupt-driven time-sharing systems.

Summary. This completes our initial description of the Model Human Processor. To recapitulate, the Model Human Processor consists of (1) a set of interconnected memories and processors and (2) a set of principles of operation. The memories and processors are grouped into three main subsystems: a perceptual system, a cognitive system, and a motor system. The most salient characteristics of the memories and processors can be summarized by the values of a few parameters: processor cycle time τ, memory capacity μ, memory decay rate δ, and

20 On the fast end, memory scanning rates go down to 25 msec/item (Sternberg, 1975, p. 225, Figures 8 and 9, lower error bar for LETTERS). Michon (1978, p. 93) summarizes the search for the "time quantum" as converging on 20~30 msec. On the slow end, silent counting, which takes about 167 msec/item (Landauer, 1962), has sometimes been taken as a minimum cognitive task. It has sometimes been argued (Hick 1952) that the subject in a choice reaction time experiment makes one choice for each bit in the set of alternatives, in which case a typical value would be 153 msec/bit (Figure 2.22). Welford (1973, in Kornblum) has proposed a theory of choice reaction in which the subject makes a series of choices, each taking 92 msec. Blumenthal (1977) reviews an impressively large number of cognitive phenomena with time constraints in the tenth of a second range.

Rate at which an item can be matched against Working Memory:		
Digits	33 [27~39] msec/item	Cavanaugh (1972)
Colors	38 msec/item	Cavanaugh (1972)
Letters	40 [24~65] msec/item	Cavanaugh (1972)
Words	47 [36~52] msec/item	Cavanaugh (1972)
Geometrical shapes	50 msec/item	Cavanaugh (1972)
Random forms	68 [42~93] msec/item	Cavanaugh (1972)
Nonsense syllables	73 msec/item	Cavanaugh (1972)
Range =	27~93 msec/item	
Rate at which four or fewer objects can be counted:		
Dot patterns	46 msec/item	Chi & Klahr (1975)
3-D shapes	94 [40~172] msec/item	Akin and Chase (1978)
Range =	40~172 msec/item	
Perceptual judgement:		
	92 msec/inspection	Welford (1973)
Choice reaction time:		
	92 msec/inspection	Welford (1973)
	153 msec/bit	Hyman (1953)
Silent counting rate:		
	167 msec/digit	Landauer (1962)

Figure 2.7. Cognitive processing rates. Selected cycle times (msec/cycle) that might be identified with the Cognitive Processor cycle time.

memory code type κ. Each of the processors has a cycle time on the order of a tenth of a second.

A model so simple does not, of course, do justice to the richness and subtlety of the human mind. But it does help us to understand, predict, and even to calculate human performance relevant to human-computer interaction. To pursue this point, and to continue our development of the Model Human Processor, we now turn to an examination of sample phenomena of human performance.

2.2. HUMAN PERFORMANCE

We have said that in order to support cognitive engineering of the human-computer interface, an applied information-processing psychology should be based on task analysis, calculation, and approximation. These qualities are important for the Model Human Processor to possess if we are to address the practical prediction of human performance. Although it might be argued that the primitive state of development in psycho- logical science effectively prevents its employment for practical engi- neering purposes, such an argument overlooks the often large amounts of uncertainty also encountered in fields of engineering based on the physical sciences. The parameters of soil composition under a hill, the wind forces during a storm, the effects of sea life and corrosion on underwater machinery, the accelerations during an earthquake—all are cases where the engineer must proceed in the face of considerable uncertainty in parameters relevant to the success of his design.

A common engineering technique for addressing such uncertainty is to settle on nominal values for the uncertain parameters representing low, high, and typical values, and to design to these. Thus a heating engineer might calculate heating load for a building at design temperatures of 10°F. for winter, 105°F. for summer, and a more common 70°F. day.

A similar technique helps us to address the uncertainties in the parameters of the Model Human Processor. We can define three versions of the model: one in which all the parameters listed are set to give the worst performance (*Slowman*), one in which they are set to give the best performance (*Fastman*), and one set for a nominal performance (*Middleman*).

The difference between the results of the Middleman (nominal) and the Fastman-Slowman (range) calculations must be kept clearly in mind. Secondary effects, outside the scope of the model, may mean that the

appropriate parameter value for a particular calculation lies at a place i the range other than that given as the nominal value: the real predic tions of the Model Human Processor are that a calculated quantity wil lie somewhere within the Slowman~Fastman range. On the other hand because these ranges are set by extreme and not particularly typica values, the range is pessimistically wide. The nominal value for each parameter allows a complement to the range calculations based on a typical value for the parameter at some increased risk of inaccuracy due to secondary effects. The two types of calculation, range and nominal can be used together in a number of ways depending on whether we are more interested, say, in assessing the sensitivity of a nominal calculation to secondary effects or in identifying the upper or lower boundary at which some user performance will occur.

We turn now to examples of human performance bearing potential relevance to human-computer interaction, relating these, where possible, to the Model Human Processor. The performances are drawn from the areas of perception, motor skill, simple decisions, learning and retrieval, and problem solving.

Perception

Many interesting perceptual phenomena derive from the fact that similar visual stimuli that occur within one Perceptual Processor cycle tend to fuse into a single coherent percept. As an example, consider the problem of the rate at which frames of a moving picture need to be changed to create the illusion of motion.

MOVING PICTURE RATE

Example 1. Compute the frame rate at which an animated image on a video display must be refreshed to give the illusion of movement.

Solution. Closely related images nearer together in time than τ_P, the cycle time of the Perceptual Processor, will be fused into a single image. The frame rate must therefore be such that:

$$\text{Frame rate} > 1/\tau_P = 1/(100 \text{ msec/frame}) = 10 \text{ frames/sec.} \blacksquare$$

This solution can be augmented by realizing that in order to be certain that the animation will not break down, the frame rate should, of course, be faster than this number. How much faster? A reasonable upper bound for how fast the rate needs to be can be found by redoing the above calculation for the Fastman version of the model (τ_P = 50 msec):

$$\begin{aligned}\text{Max frame rate for fusion} &= 1/(50 \text{ msec/frame}) \\ &= 20 \text{ frames/sec}.\end{aligned}$$

This calculation is in general accord with the frame rates commonly employed for motion picture cameras (18 frames/sec for silent and 24 frames/sec for sound).

The Model Human Processor also warns us of secondary phenomena that might affect these calculations. By the Variable Perceptual Processor Rate Principle, τ_P will be faster for the brighter screen of a cinema projector and slower for the fainter screen of a video display terminal.

MORSE CODE LISTENING RATE

Because stimuli within τ_P fuse into the same percept, the cycle time of the Perceptual Processor sets fundamental limits on the speed with which the user can attend to auditory or visual input.

Example 2. In the old type of Morse Code device, dots and dashes were made by the clicks of the armature of an electromagnet, dots being distinguished from dashes by a shorter interval between armature clicks. Subsequently, oscillators came into use which allowed the dots and dashes to be done by bleeps of different lengths. Should there be any difference between the two devices in the maximum rate at which code can be received?

Solution. With the older device, a dot requires the perception of two events (two clicks of the armatures). According to the model, this requires $2\tau_P$ msec, if each of these events is to be separately perceived. Officially a dash is defined as 3 dots in length, leading to an estimate of $6\tau_P$. However, high speed code often differs from the standard; and an expert should be able to perceive a dash as different than a dot if it is at least τ_P longer, giving $2\tau_P + \tau_P = 3\tau_P$ msec as the minimum time for a dash. Assuming a minimum $1\tau_P$ space between letters and $2\tau_P$ space between words, we can calculate the reception rate for random text by first computing the minimum reception time per letter and then weighting that by English letter frequencies, with an appropriate adjustment for word spacing. This calculation should underestimate somewhat the reception rates for each system, since it is only based on a first-order approximation to English below the word level; but it will allow a relative comparison. The probabilities for the letters in English are given in Figure 2.8 together with their Morse Code representation and the time/letter computed by the rates given above, assuming τ_P = 100 [50~200] msec. Weighting the time/code by the frequency of its occurrence gives a mean time of 709 [354~1417] msec/letter (including spacing between letters). Assuming 4.8 char/word (the value for Bryan and Harter's 1898 telegraphic speed test) gives:

$$\begin{aligned}\text{Max reception rate} &= (.709\ [.354\text{\textasciitilde}1.417] \text{ sec/letter} \\ &\qquad \times 4.8 \text{ letters/word}) \\ &\qquad + .200\ [.100\text{\textasciitilde}.400] \text{ sec/word-space} \\ &= 3.6\ [1.9\text{\textasciitilde}7.0] \text{ sec/word} \\ &= 17\ [9\text{\textasciitilde}32] \text{ words/min}.\end{aligned}$$

For the oscillator-based telegraph, on the other hand, a dot requires the perception of only one event. This should require τ_P. Assuming that a dash can be distinguished from a dot if the dash is $2\tau_P$ long, the time per letter would be 453 [227~907] msec and the calculation is:

$$\begin{aligned}\text{Max reception rate} &= (.453\ [.227\text{\textasciitilde}.907] \text{ sec/letter} \\ &\qquad \times 4.8 \text{ letters/word}) \\ &\qquad + .200\ [.100\text{\textasciitilde}.400] \text{ sec/word-space} \\ &= 2.4\ [1.3\text{\textasciitilde}4.6] \text{ sec/word} \\ &= 25\ [13\text{\textasciitilde}47] \text{ words/min}. \blacksquare\end{aligned}$$

So it would be expected that operators could receive code faster with the newer oscillator-based system than with the older system. Informal evidence suggests that this is true and that the oscillator-based rates are at least in the right vicinity. Current reception rates are faster than the rates of turn-of-the-century telegraphers, although this comparison may be confounded with the effect of sending equipment. Whereas 20~25 words/min with the old telegraph was regarded as the range for very

Letter	p	Morse Code	Calculated Minimum Reception Time: Armature System (msec)	Calculated Minimum Reception Time: Oscillator System (msec)
E	.1332	•	300 [150~600]	200 [100~400]
T	.0978	–	400 [200~800]	300 [150~600]
A	.0810	•–	600 [300~1200]	400 [200~800]
H	.0772	••••	900 [450~1800]	500 [250~1000]
O	.0663	–––	1000 [500~2000]	700 [350~1400]
S	.0607	•••	700 [350~1400]	400 [200~800]
N	.0601	–•	600 [300~1200]	400 [200~800]
R	.0589	•–•	800 [400~1600]	500 [250~1000]
I	.0515	••	500 [250~1000]	300 [150~600]
L	.0447	•–••	1000 [500~2000]	600 [300~1200]
D	.0432	–••	800 [400~1600]	500 [250~1000]
M	.0248	––	700 [350~1400]	500 [250~1000]
C	.0236	–•–•	1100 [550~2200]	700 [350~1400]
U	.0309	••–	800 [400~1600]	500 [250~1000]
W	.0287	•––	900 [450~1800]	600 [300~1200]
G	.0218	––•	900 [450~1800]	600 [300~1200]
Y	.0212	–•––	1200 [600~2400]	800 [400~1600]
F	.0179	••–•	1000 [500~2000]	600 [300~1200]
B	.0163	–•••	1000 [500~2000]	600 [300~1200]
P	.0153	•––•	1100 [550~2200]	700 [350~1400]
K	.0107	–•–	900 [450~1800]	600 [300~1200]
V	.0099	•••–	1000 [500~2000]	600 [300~1200]
J	.0015	•–––	1200 [600~2400]	800 [400~1600]
X	.0014	–••–	1100 [550~2200]	700 [350~1400]
Q	.0008	––•–	1200 [600~2400]	800 [400~1600]
Z	.0006	––••	1100 [550~2200]	700 [350~1400]

Figure 2.8. Morse codes arranged in order of frequency of individual letters.

Frequencies (as a proportion of total letters) in column p are based on Mayzner and Tresselt (1965).

good, experienced railroad telegraphers by Bryan and Harter (1898), reception rates of 45~50 words/minute are seen with the oscillator-based code (and the world record is over 75 words/minute!). This comparison is in the predicted order and, as expected, somewhat faster than our calculation based on a first-order approximation to English. A better approximation to the first-order assumptions of our calculation (but, alas, for Russian) is the set of rates achieved by a set of non-Russian-speaking telegraphers whose job it was to transliterate Russian Morse Code: 30 words/minute average, 38~40 words/minute maximum, and 45 words/minute top (Robin Kinkead, personal communication)—rates consonant with our oscillator-based calculation.

PERCEPTUAL CAUSALITY

One way for two distinct stimuli to fuse is for the first event to appear to *cause* the other.

> **Example 3.** In a graphic computer simulation of a pool game, there are many occasions upon which one ball appears to bump into another ball, causing the second one to move. What is the time available, after the collision, to compute the initial move of the second ball, before the illusion of causality breaks down?

Solution. The movements of the first and second balls must appear to be part of the same event in order for the collision to appear to cause the movement of the second ball, if the movement occurs within one cycle of 100 msec. Since the illusion will break down in the neighborhood of 100 msec, the program should try to have the computation done well before this time. The designer can be sure the illusion will hold if designed for Fastman, with the computation done in 50 msec. ∎

Figure 2.9 shows the results of an experiment analogous to Example 3 in which subjects had to classify collisions between objects (immediate causality, delayed causality, or independent events) as a function of the delay before the movement of the second object. The perception of immediate causality ends in the neighborhood of 100 msec; some degradation of immediate causality begins for some subjects as early as 50 msec.

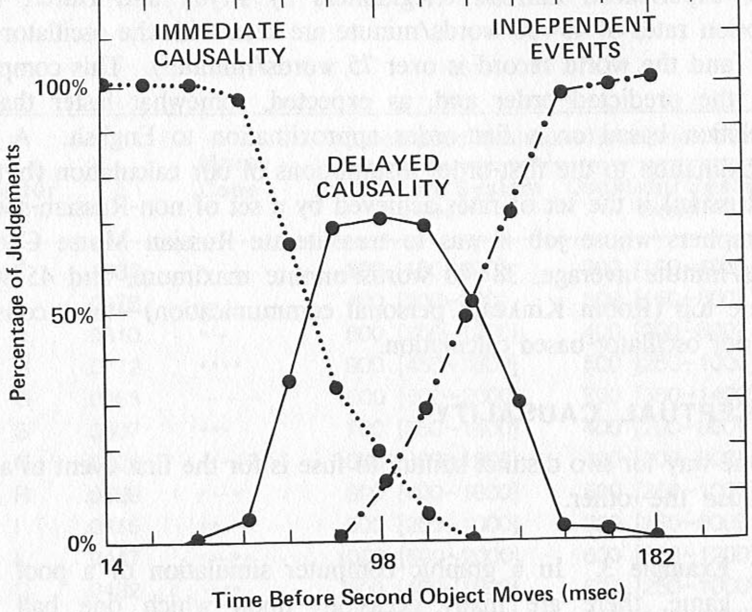

Figure 2.9. Perceived causality as a function of inter-event time between the motion of two objects.
Three types of perceived causality are shown as a function of the interval separating the end of Object A's motion and the beginning of the second object's motion. Average over three subjects. From Michotte (1963, Figure 5, p. 94).

READING RATE

Many perceptual phenomena concern a visual area large enough that the fovea of the eye must be moved to see them. When eye movements are involved, they can dominate the time required for the task.

Example 4. How fast can a person read text?

Solution. Assuming 230 msec/saccade (from Figure 2.1), a reading rate can be calculated from assumptions about how much the reader sees with each fixation. If he were to make one saccade/letter (5 letters/word), the reading rate would be:

$$(60 \text{ sec/min})/(.230 \text{ sec/saccade} \times 5 \text{ saccade/word}) = 52 \text{ words/min}.$$

For one saccade/word, the rate would be:

$$(60 \text{ sec/min})/(.230 \text{ sec/saccade} \times 1 \text{ saccade/word}) = 261 \text{ words/min}.$$

For one saccade/phrase (containing the number of characters/fixation found for good readers, 13 chars = 2.5 words), the rate would be:

$$(60 \text{ sec/min})/(.230 \text{ sec/saccade} \times 1/2.5 \text{ saccade/word}) = 652 \text{ words/min}.$$ ∎[21]

How much the reader takes in with each fixation is a function of the skill of the reader and the perceptual difficulty of the material. If the material is conceptually difficult, then the limiting factor for reading rate will not be in the eye-movement rate, but in the cognitive processing. The calculation implies that readers who claim to read much more than 600 words/min do not actually see each phrase of the text. In other words, speed readers skim.

Motor Skill

Just as fundamental limits on the rate of user perceptual performance were set by the cycle time of the Perceptual Processor, limits on movement are set by the rates of the Perceptual and Motor Processors. Two basic kinds of movement occur in human-computer interaction: (1) movement of the hand towards a target and (2) keystrokes.

FITTS'S LAW

The first kind of movement, moving the hand towards a target, can be understood, and an expression for movement time derived, using the Model Human Processor plus some assumptions.[22] Suppose a person wishes to move his hand D cm to reach an S cm wide target (see Figure 2.10). The movement of the hand, as we have said, is not continuous, but consists of a series of microcorrections, each with a certain accuracy.

21 This calculation is discussed in Hochberg (1976, p. 409).

22 This derivation is similar to that of Crossman and Goodeve (1963) and Keele (1968).

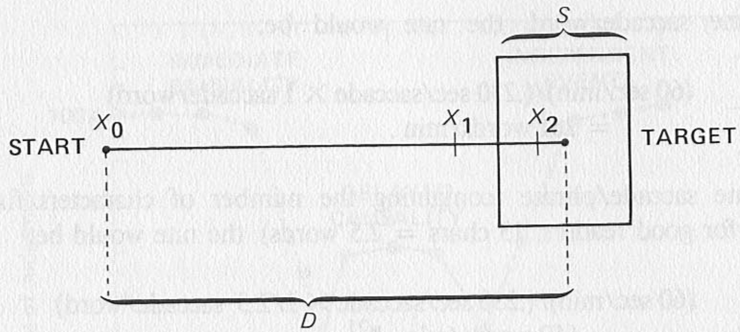

Figure 2.10. Analysis of the movement of a user's hand to a target.
The hand starts from the point labeled START and is to move to anywhere inside the TARGET as fast as possible. D is the distance to the target and S is the width of the target.

To make a correction takes at minimum one cycle of the Perceptual Processor to observe the hand, one cycle of the Cognitive Processor to decide on the correction, and one cycle of the Motor Processor to perform the correction, or $\tau_P+\tau_C+\tau_M$. The time to move the hand to the target is then the time to perform n of these corrections or $n(\tau_P+\tau_C+\tau_M)$. Since $\tau_P+\tau_C+\tau_M \simeq 240$ msec, n is the number of roughly 240-msec intervals it takes to point to the target.

Let X_i be the distance remaining to the target after the ith corrective move and X_0 (= D) be the starting point. Assume that the relative accuracy of movement is constant, that is, that $X_i/X_{i-1} = \varepsilon$, where $\varepsilon < 1$ is the constant error. On the first cycle the hand moves to

$$X_1 = \varepsilon X_0 = \varepsilon D .$$

On the second cycle, the hand moves to

$$X_2 = \varepsilon X_1 = \varepsilon(\varepsilon D) = \varepsilon^2 D .$$

On the nth cycle it moves to

$$X_n = \varepsilon^n D . \qquad (2.1)$$

The hand stops moving when it is within the target area, that is when

$$\varepsilon^n D \leq \tfrac{1}{2} S .$$

Solving for n gives

$$n = -\log_2(2D/S) / \log_2 \varepsilon .$$

Hence the total movement time T_{pos} is given by

$$T_{pos} = n(\tau_P+\tau_C+\tau_M)$$

$$T_{pos} = I_M \log_2 (2D/S), \qquad (2.2)$$
$$\text{where } I_M = -(\tau_P+\tau_C+\tau_M) / \log_2 \varepsilon .$$

Equation 2.2 is called Fitts's Law. It says that the time to move the hand to a target depends only on the relative precision required, that is, the ratio between the target's distance and its size. Figure 2.11*a* plots movement time according to Equation 2.2 for an experiment in which subjects had to alternate tapping between two targets S in. wide, D in. apart. The points fall along a straight line as predicted, except for points at low values of $\log_2(2D/S)$.

The constant ε has been found to be about .07 (see Keele, 1968; Vince, 1948), so I_M can be evaluated:

$$I_M = -240 \text{ msec} / \log_2(.07) \text{ bits}$$
$$= 63 \text{ msec/bit} .$$

A Fastman~Slowman calculation gives a range of I_M = 27~122 msec/bit. Several methods have been used to measure the correction time. One is to turn out the lights shortly after a subject starts moving his hand to a target and note the minimum light-on time that affects accuracy.[23] Another is to detect the onset of correction from trajectory acceleration changes.[24] These methods have given cycle time values in the range

23 For a discussion, see Welford (1968).

24 Carlton (1980); Langolf (1973); Langolf, Chaffin, and Foulke (1976).

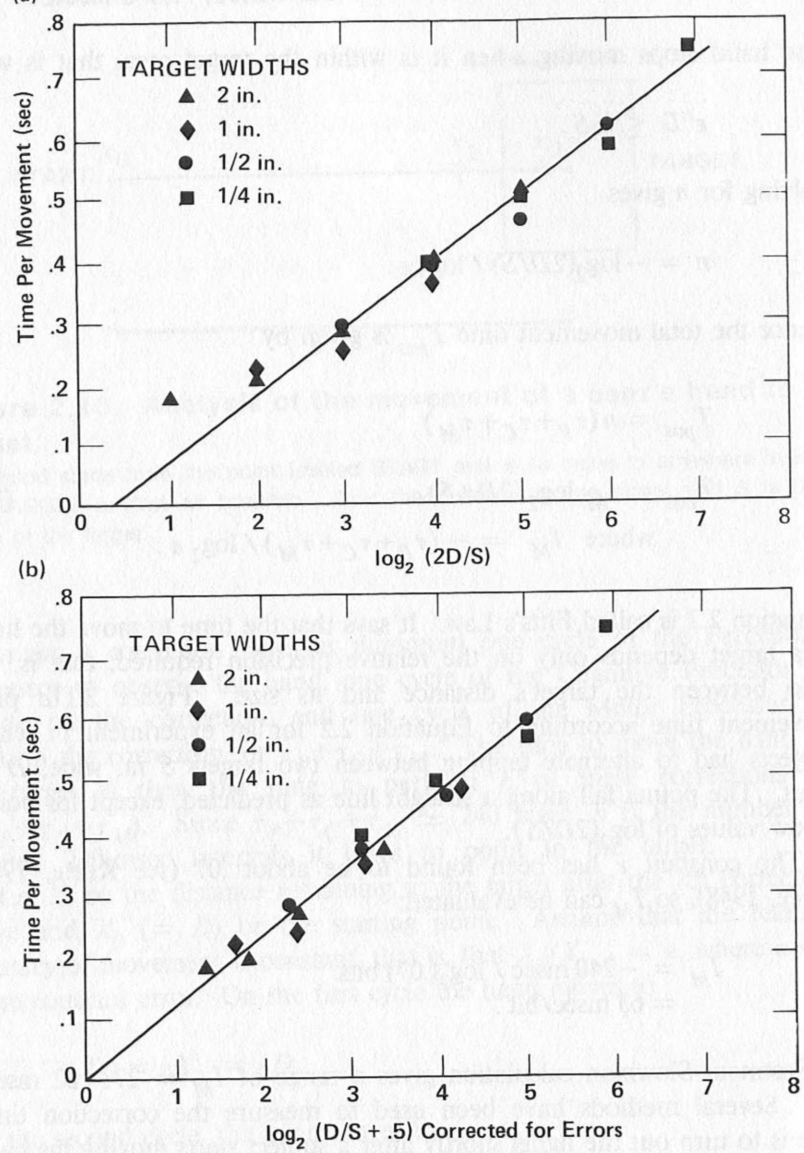

Figure 2.11. Movement time as a function of two versions of Fitts's Law.

From Welford (1968, Figures 5.3 and 5.4).

(a) Times for reciprocal tapping with a 1 oz. stylus plotted in terms of Equation 2.2. Data from an experiment by Fitts (1954). Each point is based on a total of 613~2669 movements obtained from 16 subjects.

(b) The same data as in (a) plotted in terms of Equation 2.3, corrected for errors by Crossman's method (see Welford, 1968).

$\tau_P + \tau_C + \tau_M$ = 190~260 msec/cycle (we calculated $\tau_P + \tau_C + \tau_M$ = 240 msec). The measured correction times correspond to I_M = 50~68 msec/bit (we calculated 63 [27~122] msec/bit).

Measurements of I_M determined directly by plotting observations according to Equation 2.2 give somewhat higher values centering around I_M = 100 msec/bit. The slope of the line drawn through the points in Figure 2.11*a* is about I_M = 104 msec/bit. Slopes from other experiments are in the I_M = 70~120 msec/bit range. Since I_M will be useful for later calculations, we set here a value based on several experiments:

$$I_M = 100\,[50\sim120]\ \text{msec/bit}\ .^{25}$$

This value is a refinement of the value calculated from the Model Human Processor.

The problem of the points that wander off the line for low values of $\log_2(D/S)$ and the slight curvature evident in Figure 2.11*a* can be straightened by adopting a variant of Fitts's Law developed by Welford (1968):

$$T = I_M \log_2 (D/S + .5)\ . \qquad (2.3)$$

In Figure 2.11*b* the same data are plotted using Equation 2.3 (and a method of correcting for errors). All the points now lie on the line and the slight bowing has been straightened. This equation gives a somewhat higher estimate for I_M in Figure 2.11*b*, I_M = 118 msec/bit.

[25] For single, discrete, subject-paced movements, the constant is a little less than I_M = 100 msec/bit and closer to the 50~68 msec/bit value cited above for other experimental methods and for our nominal calculation. Fitts and Peterson (1964) get 70~75 msec/bit. Fitts and Radford (1966) get a value of 78 msec/bit (12.8 bits/sec). Pierce and Karlin (1957) get maximum rates of 85 msec/bit (11.7 bits/sec) in a pointing experiment. For continuous movement, repetitive, experimenter-paced tasks, such as alternately touching two targets with a stylus or pursuit tracking, the constant is a little above I_M = 100 msec/bit. Elkind and Sprague (1961) get maximum rates of 135 msec/bit (7.4 bits/sec) for a pursuit tracking task. Fitts's original dotting experiment (Figure 2.11) gives 118 msec/bit using Equation 2.3. Welford's (1968) study using Equation 2.3 and the actual distance between the dots gives 120 msec/bit.

Example 5. On a certain pocket calculator, the heavily used gold **f** button employed to shift the meaning of the keys is located on the top row (see Figure 2.12). How much time would be saved if it were located in a more convenient position just above the numbers?

Solution. Assume that the position of the **5** button is a fair representation of where the hand is just before pressing the **f** button. From the diagram, the distance from the **5** button to the present **f** button is 2 in., to the proposed location, 1 in. The button is 1/4 in. wide. By the Equation 2.3 version of Fitts's Law, movement time is $I_M \log_2 (D/S + .5)$, where I_M is expected to be about 100 msec/bit. So the difference in times required by the two locations is

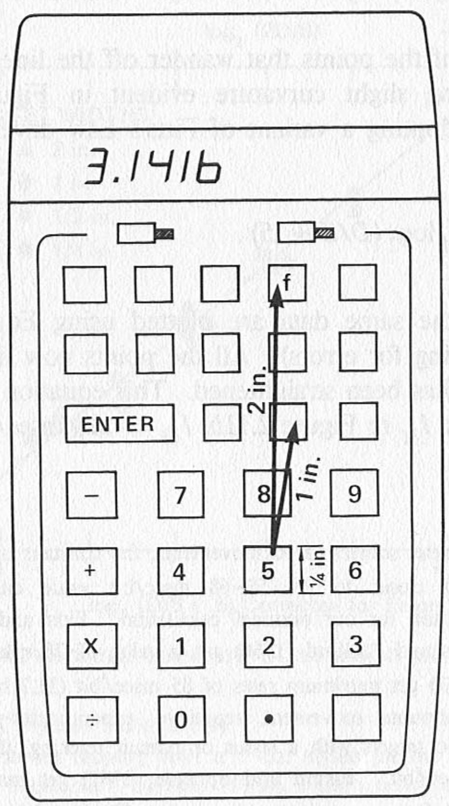

Figure 2.12. Location of keys on the pocket calculator in Example 5.

$$\begin{aligned}\Delta T &= 100\,[\log_2 (2/.25 + .5) - \log_2 (1/.25 + .5)] \\ &= 100\,(3.09 - 2.17) \\ &= 90 \text{ msec}. \blacksquare\end{aligned}$$

A test of this calculation by an informal experiment is in agreement with the predicted result. The time to press the **f** button was measured by counting the number of times the hand could alternate between the **f** and **5** button in 15 sec at both the old and the proposed location. By this method, the mean time/movement is just 15 sec/number of movements. The experiment was repeated three times:

	Old Time	*New Time*
Trial 1:	290 msec	200 msec/button-press
Trial 2:	240 msec	170 msec/button-press
Trial 3:	230 msec	180 msec/button-press
Mean:	250 msec	180 msec/button-press
Observed difference:		70 msec/button-press
Calculated difference:		90 msec/button-press

Notice that the time to press the **f** button is greater than what it could be in a more favorable location by over 1/3 (70 msec difference in a 180 msec operation). Of course, it is important to keep in mind that the design of the entire calculator will entail some trade-offs in individual key locations.

POWER LAW OF PRACTICE

Before considering the second type of motion, keystrokes, it is useful to digress to consider a learning principle applicable to perceptual-motor learning generally: The time to do a task decreases with practice. It was Snoddy (1926) who first noticed that the rate at which time improves is approximately proportional to a power of the amount of practice as given by the following relationship.

P6. Power Law of Practice. *The time T_n to perform a task on the nth trial follows a power law:*

$$T_n = T_1 n^{-\alpha} \qquad (2.4)$$

or

$$\log T_n = C - \alpha \log n \,, \tag{2.5}$$

where T_1 is the time to do the task on the first trial,
$C = \log T_1$, and α is a constant.

It can be seen in Equation 2.5 that performance time declines linearly with practice when plotted in log-log coordinates. Typical values for α are in the .2~.6 range.

Example 6. A control panel has ten keys located under ten lights. The user is to press a subset of the keys in direct response to whatever subset of lights is illuminated. If the user's response time was 1.48 sec for the 1000th trial and 1.15 sec for the 2000th trial, what is the expected response time for the 50,000th trial?

Solution. Using Equation 2.5, we can solve for T_1 in order to eliminate it.

$$T_1 = T_n n^{\alpha}$$
$$(T_{1000})1000^{\alpha} = (T_{2000})2000^{\alpha}$$
$$\alpha = \log (T_{1000}/T_{2000}) \,/\, \log (2000/1000) = .36 \,. \tag{2.6}$$

Solving for T_1 using Equation 2.6,

$$T_1 = (T_{1000})\, 1000^{.36} = 18 \text{ sec} \,.$$

The entire equation is

$$T_n = 18\, n^{-.36} \,. \tag{2.7}$$

Thus, the expected time on the 50,000th trial is

$$T_{50,000} = (18)(50,000^{-.36}) = .37 \text{ sec} \,. \blacksquare$$

Figure 2.13 shows the results of an experimental study of this situation carried out to 75,000 trials. The response time on the 50,000th trial was .40 sec compared to the .37 sec calculated. Characteristically,

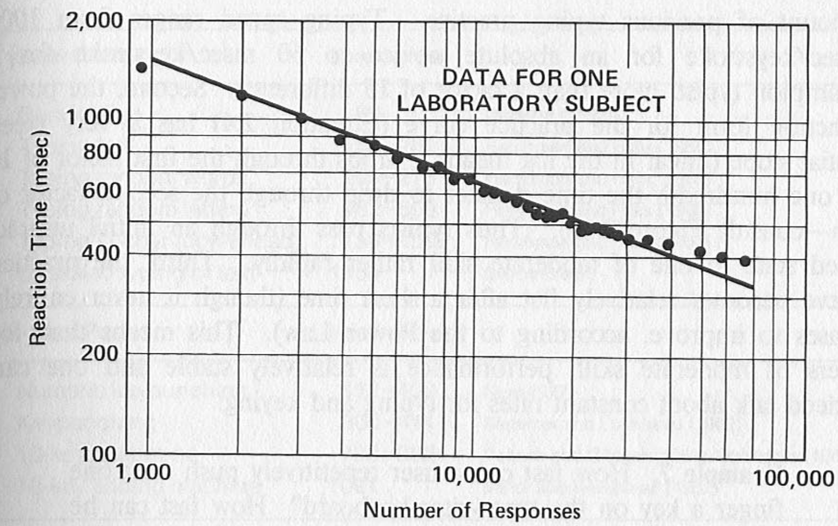

Figure 2.13. An example of the Power Law of Practice. Improvement of reaction time with practice on a 1023-choice task. Subjects pressed keys on a ten-finger chordset according to pattern of lights directly above the keys. After Klemmer (1962).

the data here are well fit by Equation 2.5, except at the ends. Estimating by eye, the best-fitting straight line in the linear portion of the curve gives $T = 21n^{-.38}$, comparable to Equation 2.7.

The Power Law of Practice applies to all skilled behavior, both cognitive and sensory-motor.[26] However, practice does not cover all aspects of learning. It does not describe the acquisition of knowledge into Long-Term Memory or apply to changes in the quality of performance. Quality does improve with practice, but it is measured on a variety of different scales, such as percentage of errors, total number of errors, and preference ratings, that admit of no uniform treatment.

KEYING RATES

The Power Law of Practice plays an important role in understanding user keystroking performance. Keying data into a system is a highly repetitive task: in a day's time, a keypuncher might strike 100,000 keys. The Power Law of Practice has three practical consequences here. First, there is a wide spread of individual differences based primarily on the

[26] See Newell and Rosenbloom (1981).

amount of previous typing practice. Typing speed ranges from 1000 msec/keystroke for an absolute novice to 60 msec/keystroke for a champion typist, more than a factor of 15 difference. Second, the power function form for the practice curve (Equation 2.4) has a very steep initial slope (linear in the log means it drops through the first factor of 10 in one hundredth the time it takes to drop through the second factor of ten—consult Figure 2.13). Thus typists pass through an initial unpracticed state to one of moderate skill rather rapidly. Third, the practice curve becomes relatively flat after a short time (though it never entirely ceases to improve, according to the Power Law). This means that, for users of moderate skill, performance is relatively stable and one can indeed talk about constant rates for typing and keying.

Example 7. How fast can a user repetitively push with one finger a key on the typewriter keyboard? How fast can he push two keys using alternate hands?

Solution. In the case of a repeated keystroke, the finger must first be cocked back, then brought forward. Each half of the stroke, according to the Model Human Processor, will take $\tau_M = 70$ msec and the whole stroke will take $\tau_M + \tau_M = 140$ msec. In the case of keystrokes between alternate hands, it should be possible for one hand to stroke while the other is cocking if the strokes are coordinated, so in these cases strokes could follow each other within 70 msec. ∎

These two are the fastest and slowest cases, hence the typing rate for a skilled typist might be expected to lie somewhere within 70~140 msec/keystroke for a mixture of same-hand and different-hand stroke combinations (if the typist is given sufficient look-ahead so that perceptual and cognitive processing overlaps motor processing).

Figure 2.14 gives data-entry rates for some keystroke-operated devices. For typewriter-like devices, expert typing rates hover in the 100~300 msec range, as expected. Champion keypunch and typing performance is in the 60~80 msec range, faster than the Middleman calculation above, but slower than the 30 msec lower bound set by a Fastman calculation. As Figure 2.14 shows, difficult text or lack of expertise exact perceptual and cognitive costs that slow the rate.

More detailed calculations of user performance can be made using data for individual interkeystroke times such as those collected by Kinkead (1975) and reproduced in Figure 2.15, which breaks down

Typewriters	(msec/stroke)	
Best keying	60	Dresslar (1892)
Typing text	158~231	Hershman and Hillix (1965)
Typing random words	200~273	Hershman and Hillix (1965)
Typing random letters	462~500	Hershman and Hillix (1965)
Typing (1 char look-ahead)	750~1500	Hershman and Hillix (1965)
Unskilled typing of text	1154	Devoe (1967)
10-Key Pads	(msec/stroke)	
Numeric keypunching	112~400	Neal(1977)
Keypunching	300~444	Klemmer and Lockhead (1962)
10-key telephone	789~952	Pollock and Gildner (1963), Deininger (1960)
10-key adding machine	1091	Minor and Revesman (1962)
Other Keyboards	(msec/stroke)	
Simple pushbuttons	570~690	Munger, Smith, and Payne (1962)
5×5 adding machine	600~800	Pollock and Gildner (1963)
Coded physician's order	779~2222	Minor and Pittman (1965)
10×10 adding machine	1200	Minor and Revesman (1962)
Chord Sets	(msec/chord)	
Stenotypists	333	Seibel (1964)
8-key chordset	508~1017	Pollock and Gildner (1963)
Mail sorting	517~882	Cornog and Craig (1965)
Hand Entry	(msec/char)	
Hand printing	545~952	Devoe (1967)
Handwriting	732	Devoe (1967)
Mark sensing	800~3750	Devoe (1967), Kolesnick and Teel (1965)
Hand punching	3093	Kolesnick and Teel (1965)

Figure 2.14. Keying times for selected input techniques.

interkeystroke times by key and by whether the preceding keystroke was on the same hand, finger, or key as the current keystroke. These times can be used to make approximate comparisons between keyboard layouts.

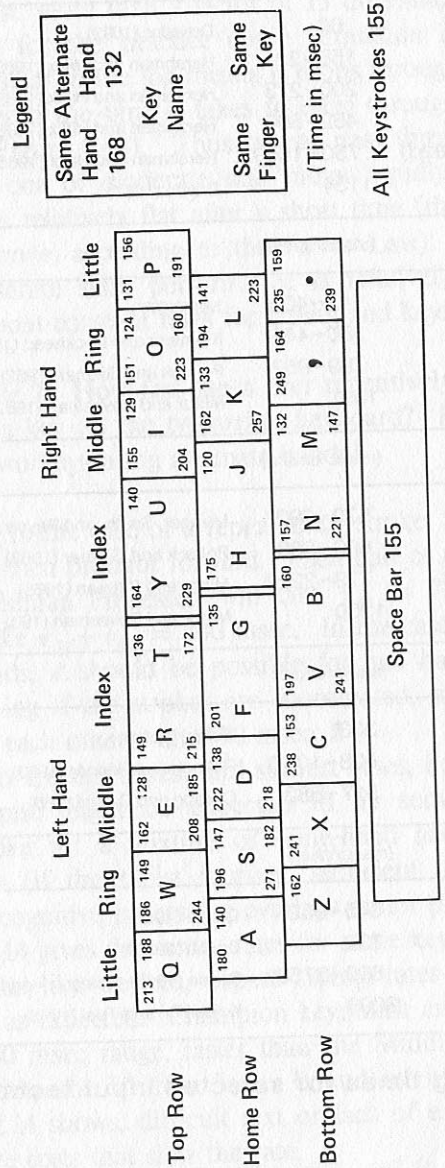

Figure 2.15. Interkeystroke typing times.
Based on 155,000 keystrokes from 22 typists (from Kinkead, 1975).

Example 8. A manufacturer is considering whether to use an alphabetic keyboard (see Figure 2.16) on his small business computer system. Among several factors influencing his decision is the question of whether experienced users will find the keyboard slower for touch-typing than the standard Sholes (QWERTY) keyboard arrangement. What is the relative typing speed for expert users on the two keyboards?

Solution. Figure 2.15 gives the time/keystroke t_i for all but the most infrequent letter keys, broken down by whether the previous key was the same key, the same finger, the same hand, or the other hand. Figure 2.17 gives the frequencies f_i with which two-letter combinations appear in English (punctuation and space digraphs are, unfortunately, not available in the table). The expected typing rate is just the weighted average,

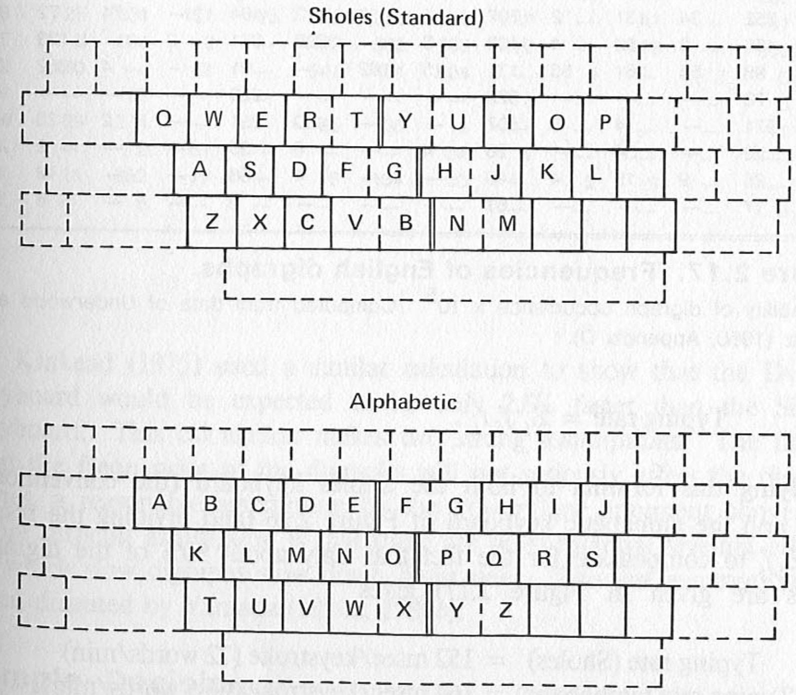

Figure 2.16. Arrangement of letter keys on Sholes and on one possible alphabetic typewriter.

First Letter	Second Letter A	B	C	D	E	F	G	H	I	J	K	L	M
A	2	229	354	242	9	115	214	13	375	19	142	842	335
B	182	15	—	2	547	—	—	—	121	13	—	227	—
C	562	—	49	—	496	—	4	543	248	—	168	125	—
D	172	—	—	36	660	8	34	6	403	—	—	51	11
E	880	13	337	1213	433	112	110	19	165	2	38	583	310
F	174	2	—	—	233	127	—	—	290	—	—	66	—
G	136	—	—	—	380	2	53	312	170	—	—	61	2
H	1056	9	—	4	3139	8	2	—	848	—	—	8	6
I	210	66	589	310	329	218	265	—	—	—	59	543	339
J	32	—	—	—	44	—	—	—	4	—	—	—	—
K	8	4	—	2	293	4	2	4	138	—	—	17	—
L	452	13	6	337	937	61	4	2	655	—	25	740	34
M	547	106	—	—	757	9	—	—	325	—	—	6	76
N	250	—	254	1476	846	36	1190	19	288	15	70	79	28
O	64	68	132	208	45	942	62	11	74	6	87	365	553
P	343	—	—	—	435	—	—	61	142	—	2	295	6
Q	—	—	—	—	—	—	—	—	—	—	—	—	—
R	577	32	108	167	1730	19	76	15	615	—	112	129	117
S	252	34	131	2	797	11	2	473	464	—	74	72	102
T	456	9	62	4	1103	8	—	3397	971	2	—	138	42
U	98	55	161	55	131	15	182	—	91	—	4	352	297
V	78	—	—	—	929	—	—	—	229	—	—	—	—
W	571	—	4	6	507	—	—	490	231	—	2	23	2
X	23	—	34	—	28	4	—	6	25	—	—	2	—
Y	25	9	15	4	140	—	—	4	38	—	—	13	28
Z	17	—	—	—	61	—	—	—	8	—	—	6	—

First Letter	Second Letter N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	2146	2	193	2	1128	1028	1362	115	252	70	13	272	25
B	2	293	—	—	140	15	4	246	4	—	—	127	—
C	—	653	—	2	333	9	333	81	—	—	—	32	—
D	34	257	4	—	108	161	2	131	21	8	—	70	—
E	1355	72	149	25	2106	1285	431	13	288	170	185	204	4
F	6	431	—	—	210	—	127	123	—	—	—	4	—
G	32	184	6	—	176	81	19	87	—	—	—	13	—
H	13	—	2	—	98	23	197	127	2	11	—	19	—
I	2394	471	68	2	386	1105	1238	8	288	—	26	—	62
J	—	89	—	—	—	—	—	57	—	—	—	—	—
K	97	—	2	—	2	59	2	—	—	—	—	15	—
L	11	378	28	—	9	112	106	100	26	25	—	481	—
M	2	386	206	—	19	78	2	142	—	—	—	114	—
N	64	486	4	8	6	384	967	87	34	—	2	134	9
O	1487	390	225	2	1239	284	466	1306	138	435	21	42	8
P	2	252	174	—	343	49	62	91	—	—	—	13	—
Q	—	—	—	—	—	—	—	115	—	—	—	—	—
R	202	819	17	—	114	458	299	134	62	8	—	252	—
S	25	331	157	23	2	386	1151	242	—	47	—	61	—
T	8	694	2	—	413	363	263	216	—	78	—	202	—
U	460	—	142	—	541	481	524	—	9	—	2	8	2
V	—	55	—	—	—	—	—	2	—	—	—	6	—
W	89	274	—	—	25	28	6	—	—	—	—	11	—
X	—	2	61	—	—	—	34	4	—	—	—	—	—
Y	11	352	17	—	6	104	30	—	2	9	—	—	—
Z	—	6	—	—	—	—	—	—	—	—	—	8	13

Figure 2.17. Frequencies of English digraphs.
Probability of digraph occurrence x 10^5. Computed from data of Underwood and Schulz (1960, Appendix D).

$$\text{Typing rate} = \Sigma_i\, f_i\, t_i\,.$$

Applying this formula to both the Sholes keyboard (the conventional one) and the alphabetic keyboard of Figure 2.16 (and dividing the result by $\Sigma_i f_i$ to compensate for the fact that only about 90% of the digraph times are given in Figure 2.17) gives

$$\text{Typing rate (Sholes)} = 152 \text{ msec/keystroke (72 words/min)}$$
$$\text{Typing rate (alphabetic)} = 164 \text{ msec/keystroke (66.5 words/min).}$$

The alphabetic arrangement is calculated to be about 8% slower than the Sholes arrangement. ∎

Kinkead (1975) used a similar calculation to show that the Dvorak keyboard would be expected to be only 2.6% faster than the Sholes keyboard. This calculation makes two strong assumptions. The first is that the frequencies of the digraphs will not seriously affect the digraph times, a reasonable assumption by the Power Law argument above. A more difficult assumption is that there are no substantial leveling effects, in which slow digraphs slow down faster ones. This last assumption has been disputed by Yamada (1980*a*, 1980*b*).

Simple Decisions

We have discussed how simple calculations are possible for perceptual and motor performance; now we can consider how the perceptual and motor systems, together with central cognitive mechanisms, combine in simple acts of behavior.

SIMPLE REACTION TIME

The basic reaction time for simple decisions can be derived from Figure 2.1.

> **Example 9.** A user sits before a computer display terminal. Whenever any symbol appears, he is to press the space bar. What is the time between signal and response?

Solution. Let us follow the course of processing through the Model Human Processor in Figure 2.1. The user is in some state of attention to the display (Figure 2.18*a*). When some physical depiction of the letter A (we denote it α) appears, it is processed by the Perceptual Processor, giving rise to a physically-coded representation of the symbol (we write it α') in the Visual Image Store and very shortly thereafter to a visually coded symbol (we write it α'') in Working Memory (Figure 2.18*b*). This process requires one Perceptual Processor cycle τ_P. The occurrence of the stimulus is connected with a response (Figure 2.18*c*), requiring one Cognitive Processor cycle, τ_C. The motor system then carries out the actual physical movement to push the key (Figure 2.18*d*), requiring one Motor Processor cycle, τ_M. Total time required is $\tau_P+\tau_C+\tau_M$. Using Middleman values, the total time required is $100+70+70=240$ msec. Using Fastman and Slowman values gives a range 105~470 msec. ∎

In practice, measured times for a simple reaction under laboratory conditions range anywhere from 100 to 400 msec.

PHYSICAL MATCHES

If the user has to compare the stimulus to some code contained in memory, the processing will take more steps.

> **Example 10.** The user is presented with two symbols, one at a time. If the second symbol is identical to the first, he is to push the key labeled YES, otherwise he is to push NO. What is the time between signal and response for the YES case?

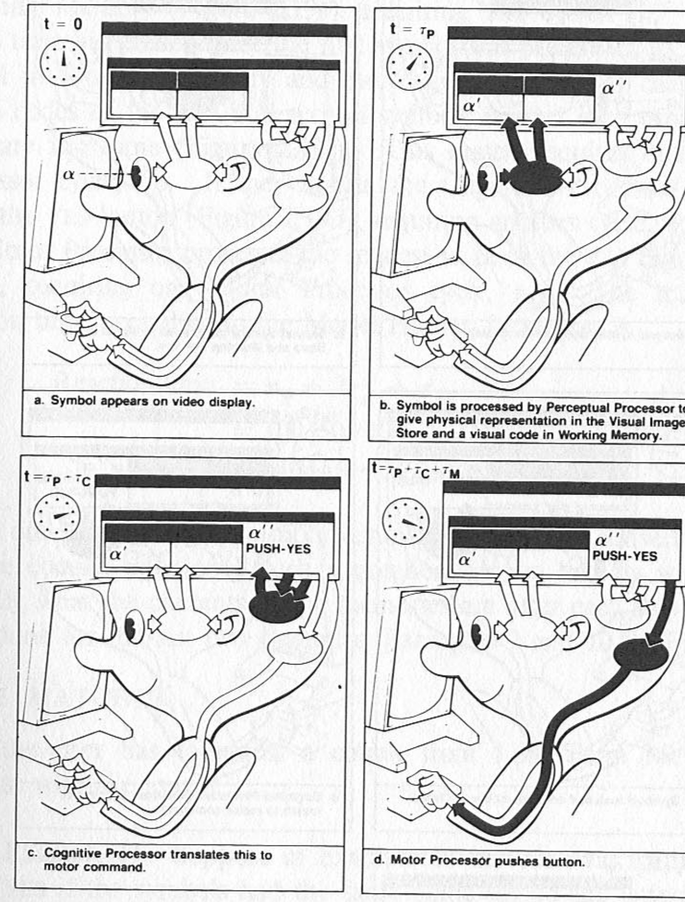

Figure 2.18. Simple reaction-time analysis using the Model Human Processor.

Solution. The first symbol is presented on the screen where it is observed by the user and processed by his Perceptual Processor, giving rise to associated representations in his Visual Image Store and Working Memory. The second symbol is now flashed on the screen and is similarly processed (Figure 2.19*a*). Since we are interested in how long it takes to respond to the second symbol, we now start the clock at 0. The Perceptual Processor processes the second symbol to get an iconic representation in Visual Image Store and then a visual representation in

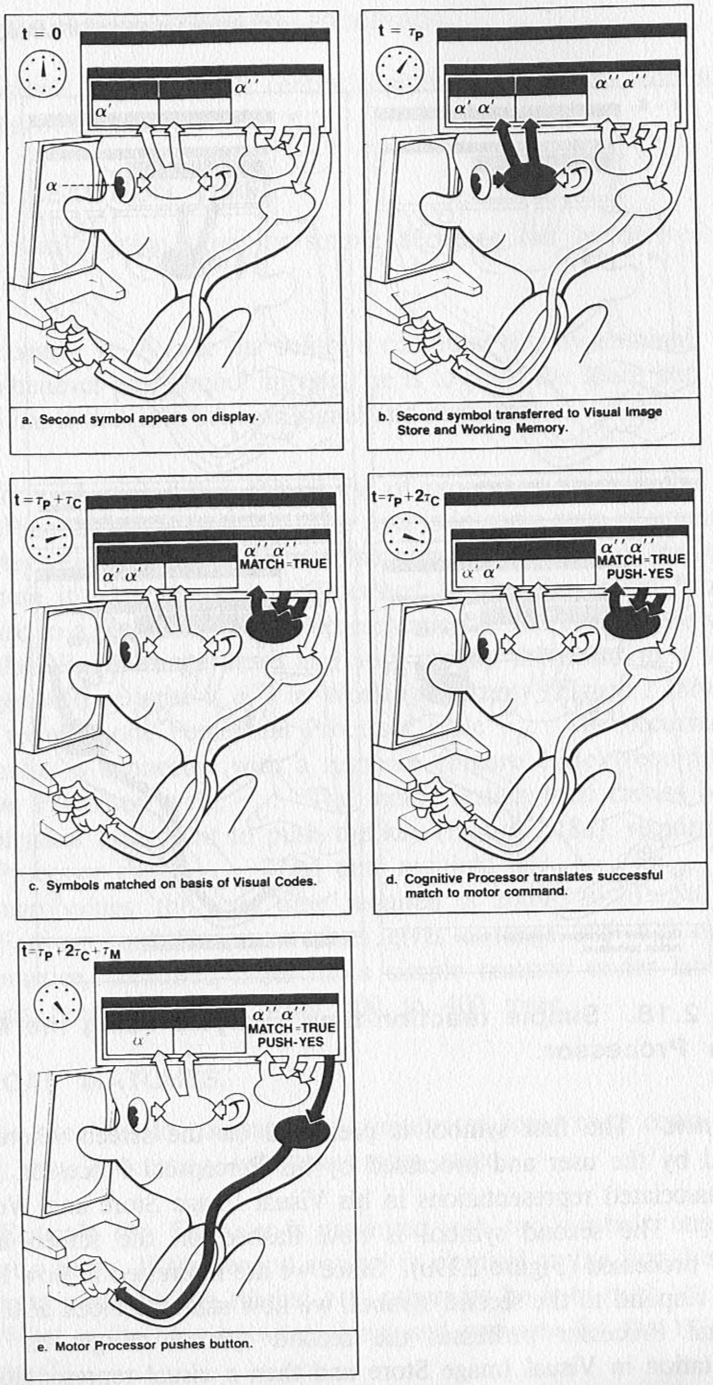

Figure 2.19. Physical name-match analysis using the Model Human Processor.

Working Memory (Figure 2.19*b*), requiring one cycle, τ_P. If not too much time has passed since the first symbol was presented, its visual code is still in Working Memory and the Cognitive Processor can match the visual codes of the first and second symbols against each other to see if they are the same (Figure 2.19*c*). This match requires one Cognitive Processor cycle, τ_C. If they match, the Cognitive Processor decides to push the YES button (Figure 2.19*d*), requiring another cycle, τ_C. Finally the Motor Processor processes the request to push the YES button (Figure 2.19*e*), requiring one Motor Processor cycle, τ_M. The total elapsed reaction time, according to the Model Human Processor, is

$$\begin{aligned}\text{Reaction time} &= \tau_P + 2\tau_C + \tau_M \\ &= 100\,[50\sim200] + 2\times(70\,[25\sim170]) + 70\,[30\sim100] \\ &= 310\,[130\sim640]\ \text{msec}.\end{aligned}$$

As our analyses become more complex, it becomes convenient to use a more concise notation. Such a notation can be had by writing symbolically what the contents of the memories are after each step. This has been done for the last two examples, Examples 9 and 10, in Figure 2.20.

NAME MATCHES

If the user has to access a chunk from Long-Term Memory, the response will take longer.

> **Example 11.** Suppose in Example 10 the user was to press YES if the symbols had the same name (as do the letters A and a), regardless of appearance and NO if they did not. What is the time between signal and response for the YES response?

The analysis is similar to the previous example except that instead of performing the match on the visual codes, the user must now wait (see Figure 2.20 Step 2.01) until the visual code has been recognized and an abstract code representing the name of the letter is available. The consequence of adding the new step is the addition of one more Cognitive Processor cycle,

$$\begin{aligned}\text{Reaction time} &= \tau_P + 3\tau_C + \tau_M \\ &= 100\,[50\sim200] + 3\times(70\,[25\sim170]) + 70\,[30\sim100] \\ &= 380\,[155\sim810]\ \text{msec}.\end{aligned}$$

Step		Display	VIS	WM	Hand	Elapsed Time
Example 9. Simple reaction						
	State at start of clock:					
1.	Symbol appears	α				0
2.	Transmitted to VIS		α'	α''		τ_P
3.	Initiate response		α'	α'', PUSH-YES		$\tau_P+\tau_C$
4.	Process motor command		α'	α'', PUSH-YES	PUSH-YES	$\tau_P+\tau_C+\tau_M$
Example 10. Physical match						
	State at start of clock:		α'	α''		
1.	Second symbol appears	α	α'	α''		0
2.	Transmitted to VIS		α', α'	α'', α''		τ_P
2.1.	*Match*		α', α'	α'', α'', MATCH = TRUE		$\tau_P+\tau_C$
3.	Initiate response		α'	α'', α'', PUSH-YES		$\tau_P+2\tau_C$
4.	Process motor command			α'', α'', PUSH-YES	PUSH-YES	$\tau_P+2\tau_C+\tau_M$
Example 11. Name match						
	State at start of clock:		α_1'	α_1'':A		
1.	Second symbol appears	α_2	α_1'	α_1'':A		0
2.	Transmitted to VIS		α_1', α_2'	α_2'', α_1'':A		τ_P
2.01.	*Recognize*		α_1', α_2'	α_2'':A, α_1'':A		$\tau_P+\tau_C$
2.1.	Match		α_1', α_2'	MATCH = TRUE		$\tau_P+2\tau_C$
3.	Initiate response		α_2'	PUSH-YES		$\tau_P+3\tau_C$
4.	Process motor command			PUSH-YES	PUSH-YES	$\tau_P+3\tau_C+\tau_M$
Example 12. Class match						
	State at start of clock:		α'	α'':A:LETTER		
1.	Second symbol appears	β	α'	α'':A:LETTER		0
2.	Transmitted to WM		α', β'	β'', α'':A:LETTER		τ_P
2.01.	Recognize		α', β'	β'':B, α'':A:LETTER		$\tau_P+\tau_C$
2.02.	*Classify*		α', β'	β'':B:LETTER, α'':A:LETTER		$\tau_P+2\tau_C$
2.1.	Match		β'	MATCH = TRUE		$\tau_P+3\tau_C$
3.	Initiate response			PUSH-YES		$\tau_P+4\tau_C$
4.	Process motor command			PUSH-YES	PUSH-YES	$\tau_P+4\tau_C+\tau_M$

Figure 2.20. Trace of the Model Human Processor's memory contents for simple decision tasks.
The symbols α and β stand for the unrecognized visual representation of the input; the symbols α' and β' stand for the physical representation of the input in the Visual Image Store (VIS); the symbols α'' and β'' stand for the visual code of the input in Working Memory (WM); and the symbols **A** and **LETTER**, stand for the abstract representation. The notation α''**:A** means that both visual and abstract codes exist in Working Memory and are associated with one another.

CLASS MATCHES

It might happen that the user has to make multiple references to Long-Term Memory.

Example 12. Suppose in Example 11 the user was to press YES if both symbols were letters, as opposed to numbers. What would be the time between signal and response?

The analysis is similar (see Figure 2.20) to the previous example except that a new step, Classify, is required to convert both versions of the symbol to the same representation.

$$\begin{aligned}\text{Reaction time} &= \tau_P + 4\tau_C + \tau_M \\ &= 100\,[50\sim200] + 4\times(70\,[25\sim170]) + 70\,[30\sim100 \\ &= 450\,[180\sim980]\text{ msec}. \blacksquare\end{aligned}$$

Experiments have been performed by many researchers to collect empirical data on the questions presented in these examples. The results are that name matches take about 70 msec longer than physical matches and that class matches take about 70 msec longer yet. (70 msec is the nominal value we have used for τ_C.) Figure 2.21 shows one such experimental result. Name matches are about 85 msec slower than physical matches when there is very little time between the first and second symbol. By the time 2 sec have elapsed, the visual code in Working Memory has decayed so that the extra step of getting the name must occur and, in fact, performance is close to that required for a name match. For these predictions, the relative, nominal value calculation gives good agreement with the data, but the absolute values of the reaction times are low (data: 525 msec, calculation: 380 [155~810] msec), reflecting some systematic, second-order effect adding a constant time to all the data points. The absolute values remain within the Fastman~Slowman range however.

CHOICE REACTION TIME

If the user has to make a choice between two responses, we can analyze the task as in Example 10 where the choices were YES and NO. If there are a larger number of choices, the situation is more complicated, but still the task can be analyzed as a sequential set of decisions made by the Cognitive Processor, each adding a nominal τ_C = 70 msec to the response.[27] Regardless of the detailed analysis of the mental steps involved in choosing between alternatives, more alternatives require more steps and, hence, more time. The relationship between time required and number of alternatives is not linear because people apparently can arrange the processing hierarchically (for example, dividing the responses into groups, then on the first cycle deciding which group should get

27 See Welford (1973) and Smith (1977).

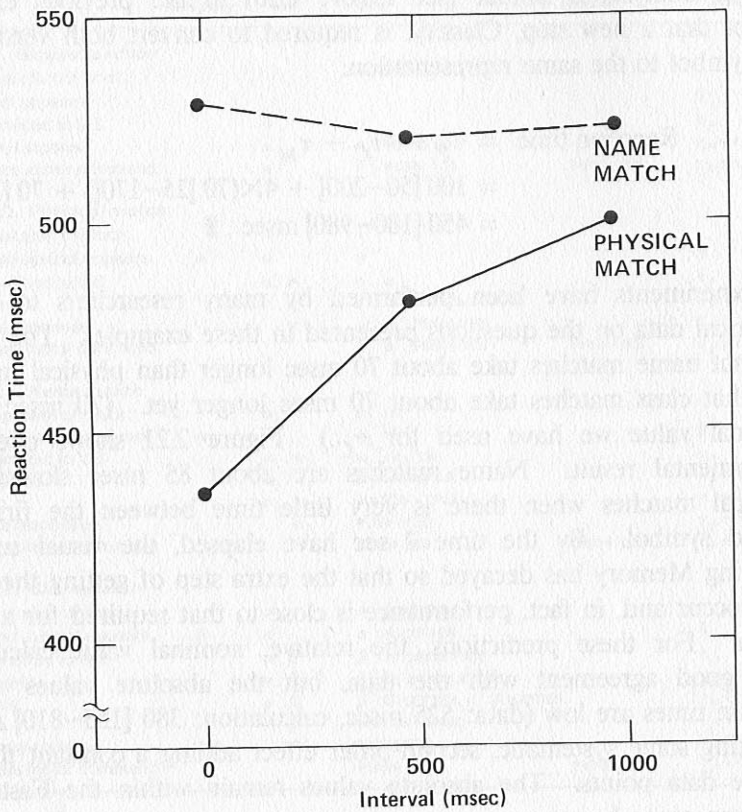

Figure 2.21. Reaction times for matching successively presented letters as a function of the inter-stimulus interval. From Posner, Boies, Eichelman, and Taylor (1969, Figure 2, p. 8).

further consideration). The minimum number of steps necessary to process the alternatives can be derived from information theory and, *to a first order of approximation,* the response time of people is proportional to the information-theoretic entropy of the decision.

P7. Uncertainty Principle: *Decision time T increases with uncertainty about the judgment or decision to be made: $T = I_C H$, where H is the information-theoretic entropy of the decision and I_C is a constant.*

For the case where a person observes n alternative stimuli, which are associated one-to-one with n responses (example: sorting multiple-part business forms by color), this principle can be given a simple mathematical formulation:

$$H = \log_2 (n + 1) . \qquad (2.8)$$

The equation, a variant of Hick's Law, may be taken as an empirical relationship that simply fits many measured situations, in that no particular mechanism is proposed. However, the equation is clearly related to rational ways of processing that minimize expected time. H is a function of $n+1$ rather than just n because there is uncertainty about whether to respond or not, as well as about which response to make. As an illustration, Figure 2.22 shows the reaction time required between the onset of one of n equally probable signals and the pressing of the appropriate button. The figure plots the reaction time against the

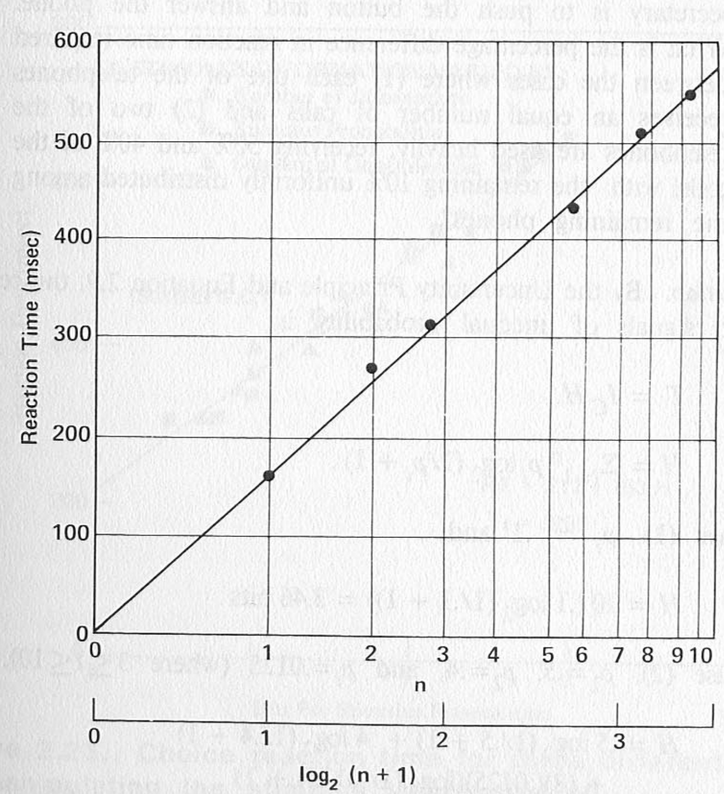

Figure 2.22. Hick's Law of choice reaction time. After Welford (1968, p. 62). At the onset of one of n lights, arranged in a row, the subject is to press the key located below the light.

number of alternatives (1 to 10), on a log scale showing that the measurements form the straight line predicted from the equation.

Equation 2.8 can be generalized to the case where the n alternatives have different probabilities of occurring,

$$H = \Sigma_{i=1}{}^{n}\, p_i \log_2 (1/p_i + 1)\,. \tag{2.9}$$

Although the probability in the formula is the person's subjective probability, it often can be estimated from the task. When all of the probabilities are equal ($= 1/n$), $p_i \log (1/p_i + 1) = (1/n) \log_2 (n+1)$ and Equation 2.9 reduces to Equation 2.8.

Example 13. A telephone call director has 10 buttons. When the light behind one of the buttons comes on, the secretary is to push the button and answer the phone. What is the percentage difference in reaction time required between the cases where (1) each one of the telephones receives an equal number of calls and (2) two of the telephones are used heavily, receiving 50% and 40% of the calls, with the remaining 10% uniformly distributed among the remaining phones?

Solution. By the Uncertainty Principle and Equation 2.9, the reaction time to signals of unequal probability is

$$T = I_C\, H,$$

where

$$H = \Sigma_{i=1}{}^{n}\, p_i \log_2 (1/p_i + 1)\,.$$

For case (1), $p_i = .1$ and

$$H = 10\,(.1 \log_2 (1/.1 + 1)) = 3.46 \text{ bits}\,.$$

For case (2), $p_1 = .5$, $p_2 = .4$, and $p_i = .0125$ (where $3 \le i \le 10$),

$$\begin{aligned} H &= .5 \log_2 (1/.5 + 1) + .4 \log_2 (1/.4 + 1) \\ &\quad + (8)(.0125)(\log_2 (1/.0125) + 1) \\ &= 2.14 \text{ bits}\,. \end{aligned}$$

The difference is $\Delta H = 3.46 - 2.14 = 1.32$ bits. So the response time for case (2) is calculated to be $2.14/3.46 = 62\%$ of the reaction time for case (1). ∎

Example 13 discussed one form of weighted occurrence probability. Another way of creating uncertainty is not to have signals occurring with fixed frequencies, but to have sequential dependencies of the signals. For instance, suppose at each trial either the signal for response #1 or response #2 can occur. However, the signal for response #1 occurs with .8 probability after a previous signal for response #1, but only with .2 probability after a signal for response #2. One can apply the same information-theoretic formula to compute the uncertainty. Hyman (1953) tried these different ways of inducing uncertainty, with the results shown in Figure 2.23. As can be seen, all the different ways of inducing uncertainty fit the same curve.

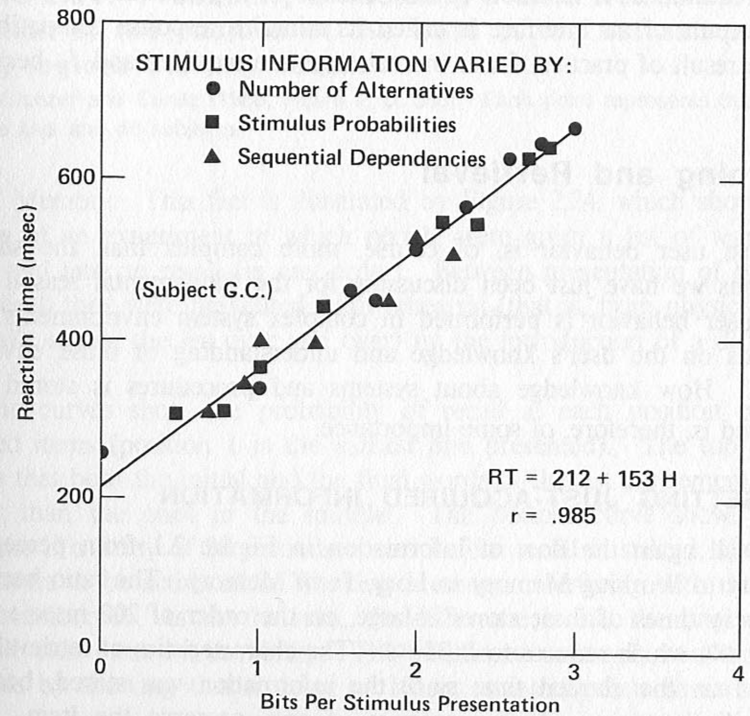

Figure 2.23. Choice reaction time for three different ways of manipulating the stimulus information *H*.
Data for a single subject. Hyman (1953, Figure 1, p. 192, subject G.C.).

Figure 2.23 shows that it takes about $I_C = 150$ msec/bit of uncertainty, above a base of about $C = 200$ ms, which we could identify as $C = \tau_P + \tau_M$. Using these values we can estimate the actual reaction times in Example 13: (1) Where each of the telephones receives an equal number of calls, the reaction time would be 200 msec + (150 msec/bit)(3.46 bits) = 719 msec. (2) Where two of the telephones are heavily used, the reaction time would be 521 msec. When the 200 msec intercept is taken into account, case (2) is 72% of case (1).

There are also situations in which we do not know how to compute H, but in which we do know that relatively more mental steps must be involved in one case than in another. For example, if the lights and keys in Example 13 were paired randomly with each other, the user would require more mental steps, I_C would be increased, and the response could be expected to take more time. The relative number of mental steps required as a function of the features of a particular set of inputs and outputs of an interface is called its stimulus-response compatibility. As the result of practice, fewer mental steps are required and I_C becomes smaller.

Learning and Retrieval

Most user behavior is, of course, more complex than the simple decisions we have just been discussing for the fundamental reason that most user behavior is performed in complex system environments and depends on the user's knowledge and understanding of those environments. How knowledge about systems and procedures is stored and retrieved is, therefore, of some importance.

FORGETTING JUST-ACQUIRED INFORMATION

Recall again the flow of information in Figure 2.1 from perceptual memory to Working Memory to Long-Term Memory. The ratio between the decay times of these stores is large, on the order of 200 msec : 7000 msec : ∞, which reduces to 1:35:∞. The characteristics of retrieval will depend on the elapsed time since the information was stored, because that will determine which memories, if any, preserve the item. For retrievals done a few seconds after input, items may be stored in either Working Memory or Long-Term Memory, or in both. For retrievals done a few minutes after input, items are retrievable only from Long-

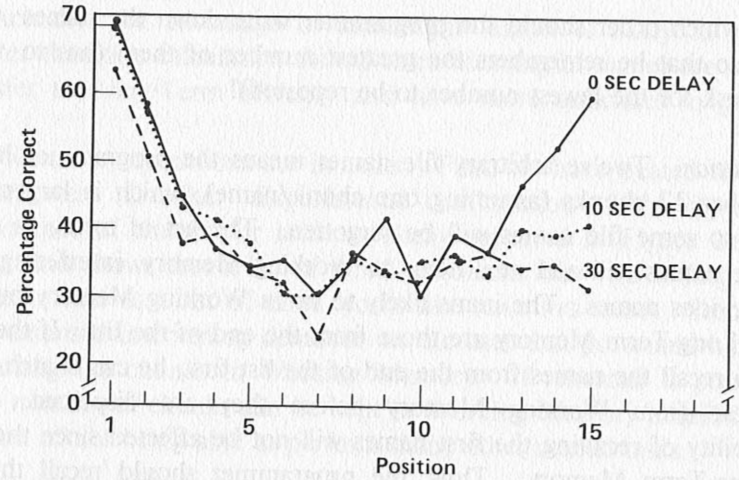

Figure 2.24. Probability of recalling a word from a list as a function of the position of the word in the list and of the delay before starting recall.
From Glanzer and Cunitz (1966, Figure 2, p. 358). Each point represents the mean for five lists and 46 subjects.

Term Memory. This fact is illustrated by Figure 2.24, which shows the results of an experiment in which people were given a list of words to learn and later to recall (in any order). Between presentation of the list and recall they were prevented from rehearsal (that is, from physically or mentally saying the list over and over) by the introduction of a different task.

The curves show the probability of recall at each position of the studied items (position 1 is the earliest one presented). The top curve shows that both the initial and the final words in the list are remembered better than the ones in the middle. The bottom curve shows what happens if a delay of 30 seconds occurs before recall is started, allowing new items to be activated in Working Memory, interfering with those to be remembered. As can be seen, the difference is that the final words lose all their extra memorability. The middle curve simply confirms the analysis by showing that a delay of 10 sec is intermediate in its effect.

Example 14. A programmer is told verbally the one-syllable file names of a dozen files to load into his programming system. Assuming the names are all arbitrary, in which order should the programmer write down the names so that he remembers the greatest number of them (has to ask for the fewest number to be repeated)?

Solution. Twelve arbitrary file names means the programmer has to remember 12 chunks (assuming one chunk/name), which is larger than μ_{WM}, so some file names will be forgotten. The act of trying to recall the file names will add new items to Working Memory, interfering with the previous names. The items likely to be in Working Memory but not yet in Long-Term Memory are those from the end of the list. If the user tries to recall the names from the end of the list first, he can snatch some of these from Working Memory before they are displaced. The probability of recalling the first names will not be affected since they are in Long-Term Memory. Thus, the programmer should recall the last names first, then the others. ∎

Example 15. Suppose that in Example 14, the 12 files did not have arbitrary names, but rather names such as **INIT1**, **INIT2**, **INIT3**, **INIT4**, **PERF1**, **PERF2**, **PERF3**, **PERF4**, **SYSTEMS1**, **SYSTEMS2**, **SYSTEMS3**, **SYSTEMS4**. In which order should the programmer write down the file names so that he remembers the largest number of them?

Solution. Unlike the case in Example 14 where each file was a separate chunk, here there are only 4 chunks: **INIT#**, **PERF#**, **SYSTEMS#**, and the rule for #. The number of chunks is within the user's Working Memory span and hence the order of recalling the files should make little difference. ∎

Example 16. Show that the amount of time a programmer can delay typing the name of the file before forgetting it (with probability > .5) is much longer if the file name is **CAT** than if it is **TXD**. (Assume the work involved does not permit the user to rehearse the file name.)

Solution. The file name **TXD** is assumed to be a nonsense word and therefore must be coded in three chunks. From Figure 2.1, δ_{WM}(3 chunks) = 7 [5~34] sec, but the file name **CAT** is one chunk, δ_{WM}(1 chunk) = 73 [73~226] sec. Nominally, the user can remember the meaningful name on the order of 73 sec / 7 sec = 10 times longer. ∎

Actually, the advantage of meaningful names is likely to be even greater than this calculation shows, since meaningful names are easier to transfer to Long-Term Memory and have more associates to get them back.

Two more comments are in order. First, we have treated chunks as if they were all alike. Experimental confirmation of the *approximate* equivalence of chunks for memory decay appeared in Figure 2.6. The figure thus shows that a list of three consonants like **TXD** is forgotten at the same rate as a list of three words like (**CAT PIG MAN**). Second, we have assumed intervening demands on the user that prevented him from rehearsing the chunks in Working Memory. If rehearsal is possible, a small number of chunks can be kept in Working Memory indefinitely, at the cost of not being able to perform many other mental tasks.

INTERFERENCE IN WORKING MEMORY

According to the Discrimination Principle, it is more difficult to recall an item if there are other similar items in memory. The similarity between two items in memory depends on the mental representation of each item, which depends in turn on the memory in which the item resides. The two most important dimensions of interference are acoustic interference and semantic interference. Items in Working Memory are usually more sensitive to acoustic interference (they are confused with other items that sound alike) because they usually (but not necessarily) use κ = acoustic coding (Conrad, 1964). Items in Long-Term Memory are more sensitive to semantic interference (they are confused with other items with similar meaning) because they use κ = semantic coding.

Example 17. A set of error indicators in a system have been assigned meaningful three-letter words as mnemonics. The idea is that, since each word is a single chunk, more codes can be remembered and written down at a glance, and since each code is only three letters the codes will be fast to write. When the system crashes, the operator is to write down a set of up to five code words that appear in a special alphanumeric display. Which is more important to avoid (in order to minimize transcription errors), codes that are similar in sound or codes that are similar in meaning?

Because the codes are to be written down immediately, the codes will be held largely in Working Memory during transcription. Because

	Experiment I (Spoken)				Experiment III (Visual)	
	Group A (*N* = 20)		Group S (*N* = 21)		Group AV (*N* = 10)	
	Acoustically Similar	Control	Semantically Similar	Control	Acoustically Similar	Control
Word Set	*mad, man, mat, map, cad, can, cat, cap*	*cow, day, far, few, hot, pen, sup, pit*	*big, long broad, great, high, tall, large,wide*	*old, deep, foul,late, safe, hot, strong, thin*	Same as Expt. I plus *cab, max*	Same as Expt. I plus *rig,day*
Percentage Correctly Recalled	10%	82%	65%	71%	2%	58%

Figure 2.25. Acoustic vs. semantic interference in Working Memory.
Subjects studied 25 five-word lists. The words in the lists were either acoustically similar, semantically similar, or unrelated (control condition). The numbers in the table are the proportion of lists recalled entirely correctly and in the proper order. Data of Baddeley (1966) as presented in Calfee (1975, Figure 17.6).

Working Memory uses largely acoustic coding, transcription errors will occur mainly from interference between acoustic codes. Similar sounding codes should therefore be avoided. ∎

Figure 2.25 shows the result of a similar experiment in which subjects had to remember lists of five words, then recall them twenty seconds later. They made many errors with the acoustically similar lists (only 1~2% of the lists were recalled error-free), but substantially fewer with the semantically similar lists (13% of the lists were recalled error-free), and this was true regardless of whether they were given the lists aurally or visually.

INTERFERENCE IN LONG-TERM MEMORY

The Discrimination Principle P4 says that the difficulty of recall depends on what other items can be retrieved by the same cues. Thus, as the user accumulates new chunks in Long-Term Memory, old chunks that are semantically similar to the new chunks become more difficult to remember.

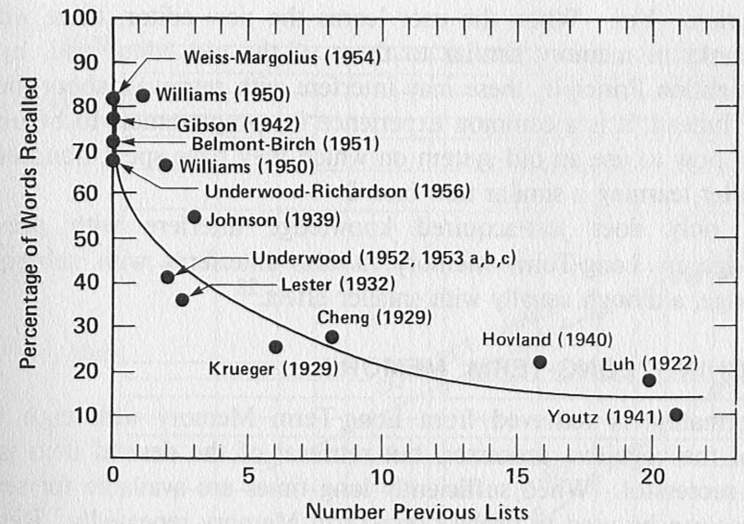

Figure 2.26. Interference of previously learned material with later learning.
Recall of serial lists 24 hours later as a function of number of previous lists learned. Revised version of Underwood (1957, Figure 3, p. 53).

A demonstration of this fact is shown in Figure 2.26. When people learn lists of words in the laboratory, they forget a large fraction of them within 24 hours. Underwood (1957) managed to find 16 separate published studies that both recorded the amount of forgetting after 24 hours and gave enough detail to determine the number of previous lists that had been learned prior to the one tested. Even though these lists differed in length, time per list item, and details of experimental procedure, it is clear that learning more prior lists results in more forgetting and that this accounts for a very large fraction of the forgetting that occurs. The size of the interference effect shows that much of what passes for forgetting is failure to retrieve, not actual loss from the memory.

Example 18. A user is about to learn how to use a new, line-oriented text-editor, identical to one he already knows except for the command names (such as ERASE instead of DELETE). Will his learning of the new editor interfere with his ability to remember the command names of the old one?

Solution. Yes. When the user learns the new editor, there will be new chunks in memory similar to those of the old editor and, by the Discrimination Principle, these may interfere with retrievals about the old editor. Indeed, it is a common experience for programmers to be unable to recall how to use an old system on which they have spent hundreds of hours after learning a similar new one. ∎

Not only does just-acquired knowledge interfere with previous knowledge in Long-Term Memory, it also interferes with subsequent knowledge, although usually with smaller effect.[28]

SEARCHING LONG-TERM MEMORY

Information is retrieved from Long-Term Memory with each basic cycle of the cognitive processor, but retrieval of the desired item is not always successful. When sufficiently long times are available for search, strategies can be used to probe Long-Term Memory repeatedly. Retrieving the name for a known but rarely used command is a typical example.

It is worth emphasizing the difficulty faced by the user attempting to retrieve an item from his Long-Term Memory, as given by the Encoding Specificity Principle. When he learned the item, it was encoded in some way. This encoding included various possible cues for recalling the item. At retrieval time, the user knows neither the desired item nor its recall cues. He must therefore guess, placing cues in Working Memory where they will serve as calls on Long-Term Memory on the next cycle. The guesses may be good and succeed immediately or, even if they fail, may retrieve some information that can help on a subsequent try.

A graphic example of Long-Term Memory search, emphasizing its capacity, the requirement for interactive strategic search, and the fact that Long-Term Memory is in many ways an *external* body of knowledge, like a phone book or an encyclopedia, is shown in Figure 2.27. The subject was asked, seven years after being graduated, to remember the names of all 600 members of her high school graduating class. (The experimenter had the year book.) As the graph shows, even after ten hours of trying, the subject was still retrieving new information from Long-Term Memory. Her strategy was an elaborate version of the interactive retrieval strategy above: In her mind, the subject scanned for faces, attended old parties, worked the alphabet, wandered down familiar streets asking for the house occupants. The process also produced fabrications

28 Murdock (1963).

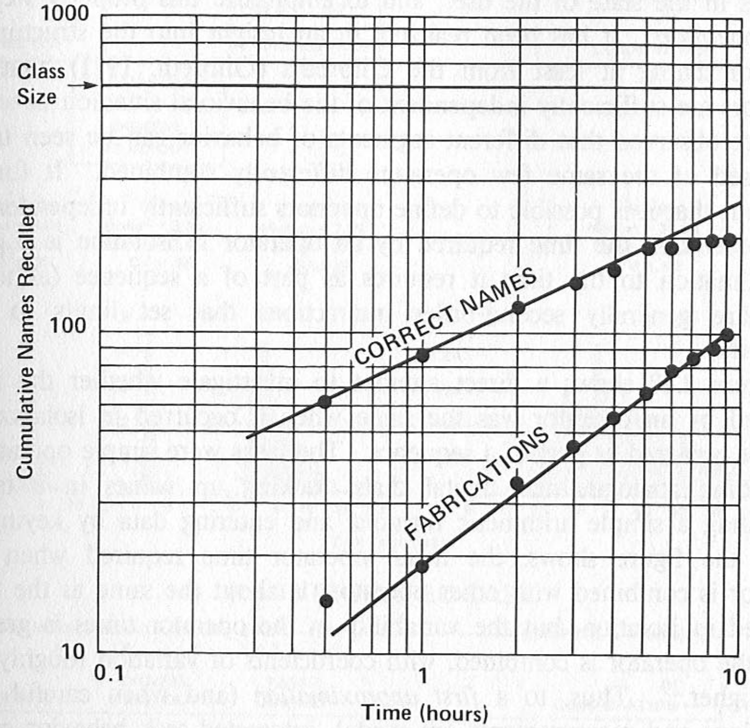

Figure 2.27. Recall of the names of high-school graduating class, seven years after being graduated.
Replotted data from Subject S1 in Williams and Hollan (1981).

where non-classmate names were recalled somewhat uncertainly during early sessions and were later misrecalled as classmate names.

Complex Information-Processing

The psychological phenomena we have discussed so far comprise the building blocks out of which more complex user behavior is composed. This more complex behavior spans longer times and is rationally organized.

OPERATOR SEQUENCES

More complex activities must ultimately be composed of the sorts of elementary actions we have been discussing. These rudimentary actions operate to cause physical changes in the state of the world or mental changes in the state of the user, and to emphasize this property we call them *operators*. It has been realized, in an insight into the structure of behavior dating at least from the Gilbreths (Gilbreth, 1911), that the operators are sufficiently independent of the behavioral situation in which they are observed that different segments of behavior can be seen to be composed of the same few operators differently combined. It further turns out that it is possible to define operators sufficiently independent of each other that the time required by an operator in isolation is a good approximation to the time it requires as part of a sequence (although there are generally second-order interactions that set limits to this additivity).

Figure 2.28 shows a direct attempt to investigate whether the time required by an operator was the same when it occurred in isolation as when it occurred as part of a sequence. The tasks were simple operations of reading analogue and digital dials, looking up values in a table, computing a simple arithmetic formula, and entering data by keying it.

As the figure shows, the mean operator time required when the operator is combined with other operators is about the same as the time required in isolation, but the variability in the operator times is greater when the operator is combined, with coefficients of variation roughly 15-20% higher.[29] Thus, to a *first approximation* (and when careful task definitions and measurements are made), integrated task behavior could be decomposed, in this case, into component operators, which could be defined and measured in independent contexts.

Example 19. In the experiment reported in Figure 2.28*b*, the total time to do the combined task was 51.56 sec (SD = 18.85). How close is this result to the times predicted from Figure 2.28*a*?

The total time to do the combined task should be the sum of the mean times for the individual tasks:

$$T = 6.24 + 3.45 + 9.26 + 34.20$$
$$= 53.15 \text{ sec}.$$

29 It is convenient to express variability in terms of the coefficient of variation CV = Standard Deviation / Mean, because it makes variability from distributions with different means more easily comparable; we often use this statistic in preference to the standard deviation.

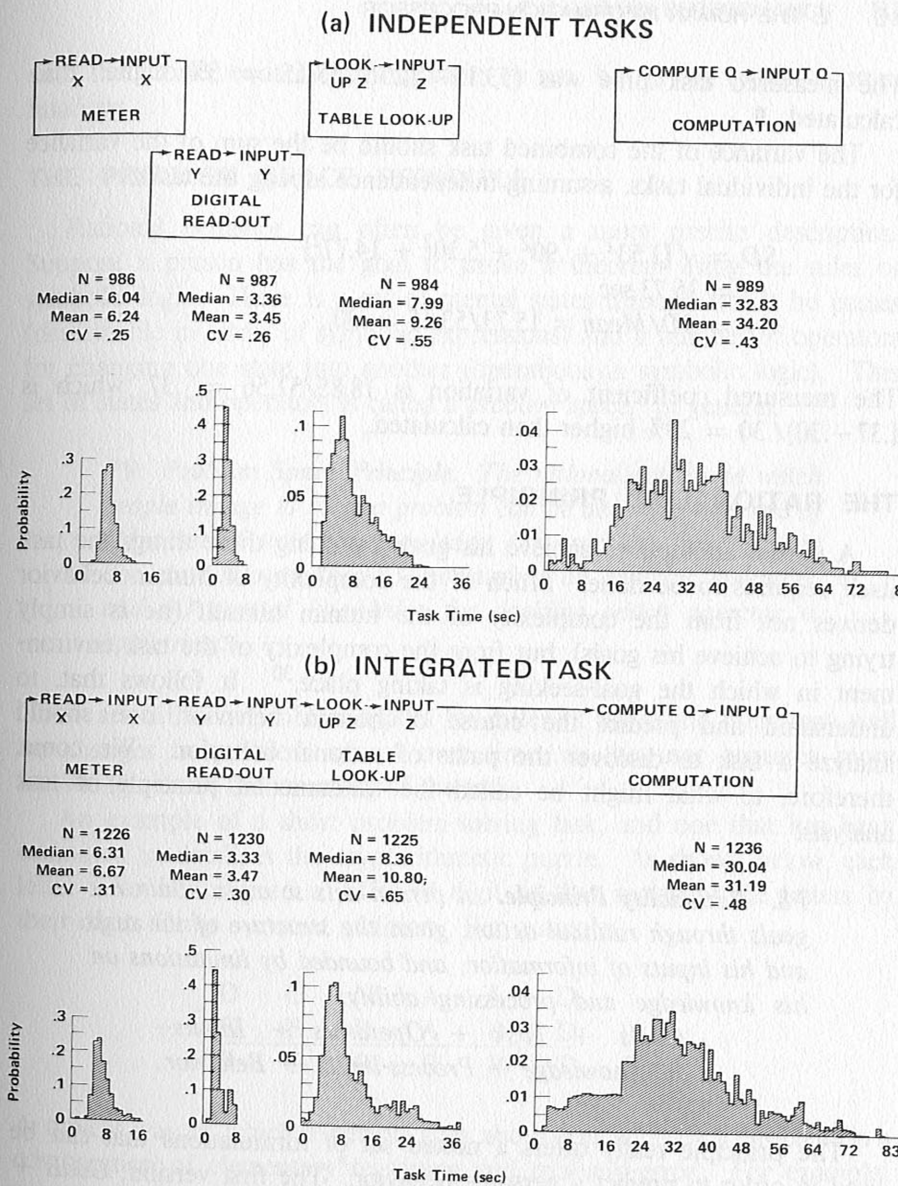

Figure 2.28. Time distributions for four operators (a) when measured in isolation and (b) when measured as part of an integrated task.

Five university students performed each of the following operators: READ-METER-AND-TYPE-INPUT, READ-DIGITAL-DISPLAY-AND-TYPE-INPUT, READ-X-Y-AND-LOOKUP-Z, READ-X-Y-Z-AND-COMPUTE-Q. They performed the operators both in isolation and as part of a larger integrated task. From Mills and Hatfield (1974, Figures 3 and 4).

The measured task time was (53.15 − 51.56)/53.15 = 3% higher than calculated. ▮

The variance of the combined task should be the sum of the variance for the individual tasks, assuming independence among the tasks:

$$SD = \sqrt{[1.53^2 + .90^2 + 5.10^2 + 14.77^2]}$$
$$= 15.73 \text{ sec}$$
$$CV = SD/Mean = 15.73/53.15 = .30\,.$$

The measured coefficient of variation is 18.85/51.56 = .37, which is (.37 − .30)/.30 = 23% higher than calculated.

THE RATIONALITY PRINCIPLE

A person attempts to achieve his goals by doing those things the task itself requires to be done. Much of the complexity of human behavior derives not from the complexity of the human himself (he is simply trying to achieve his goals), but from the complexity of the task environment in which the goal-seeking is taking place.[30] It follows that, to understand and predict the course of human behavior, one should analyze a task to discover the paths of rational behavior. We come, therefore, to what might be called the fundamental principle of task analysis:

> ***P8. Rationality Principle.*** *A person acts so as to attain his goals through rational action, given the structure of the task and his inputs of information, and bounded by limitations on his knowledge and processing ability:*
>
> $$\textit{Goals} + \textit{Task} + \textit{Operators} + \textit{Inputs}$$
> $$+ \textit{Knowledge} + \textit{Process-limits} \rightarrow \textit{Behavior}.$$

The principle really offers a nested set of formulations that can be used in order to predict a person's behavior. The first version, *Goals* + *Task* + *Operators*, takes into account only the objective situation; the other factors reflect hidden constraints, namely what the person can perceive, what he knows, and, finally, how he can compute. The additional factors offer successive approximations to how he will behave,

30 See Simon (1947, 1969), Newell and Simon (1972).

with the shorter equations being easier to use, but giving cruder approximations.

THE PROBLEM SPACE PRINCIPLE

Rational behavior can often be given a more precise description. Suppose a person has the goal to prove a theorem using the rules of symbolic logic. There is a set of mental states through which he passes (describable in terms of symbolic expressions) and a number of operators for changing one state into another (operations in symbolic logic). This set of states and operators is called a *problem space*. In general:

> ***P9. Problem Space Principle.*** *The rational activity in which people engage to solve a problem can be described in terms of (1) a set of states of knowledge, (2) operators for changing one state into another, (3) constraints on applying operators, and (4) control knowledge for deciding which operator to apply next.*

There are different problem spaces for different tasks, and there may well be changes in problem spaces over time, as the user acquires more knowledge about the structure of the task.

An example of a short problem-solving task, and one that has been examined in detail, is the cryptarithmetic puzzle. As shown below, each letter is to be assigned a different digit so that replacing the letters by their digits forms a correct addition. For example:

$$\begin{array}{rcccccc} & D & O & N & A & L & D \\ + & G & E & R & A & L & D \\ \hline & R & O & B & E & R & T \end{array} \qquad D=5$$

A typical way in which a person goes about solving such a problem is a combination of elementary reasoning and trial-and-error. For example:

> ...I can, looking at the two D's (pause) each D is 5; therefore T is 0. So I think I'll start by writing that problem here. I'll write 5, 5 is O. Now do I have any other T's? No. But I have another D. That means I have a 5 over the other side. Now I have 2 A's and 2 L's that are each somewhere and this R, 3 R's, 2L's equal and R. Of course I'm carrying a 1. Which will mean that R has to be an odd number.

```
   D O N A L D
 + G E R A L D      D = 5
 -------------
   R O B E R T
```

Informal Description: Letters in the above array are to be replaced by numerals from zero though nine, so that all instances of the same letter are replaced by the same numeral. Different letters are to be replaced by different numbers. The resulting array is to be a correctly worked problem in arithmetic. The assignment for the letter D is already given to be 5.

States: Assignments of numbers to letters.

Operators: (ASSIGN Letter Number)
(PROCESS-COLUMN Column)
(GENERATE-DIGITS Letter)
(TEST-DIGIT Number)

Path Constraint: D + D = T, etc.

Figure 2.29. External problem space for a cryptarithmetic task.

Because the 2 L's, any two numbers added together has to be an even number and 1 will be an odd number. So R can be 1... [Excerpt from protocol for Subject S3, Newell and Simon, 1972, p. 230].

The problem space for this subject (see Figure 2.29) consists of assignments of numbers to letters (**R = 3**), and various relations that can be known about the letters and digits (**R > 5, R odd, R unassigned**). The mental operators used by this subject can be identified:

ASSIGN	Assign a number to a letter.
PROCESS-COLUMN	Infer other assignments and constraints from a column.

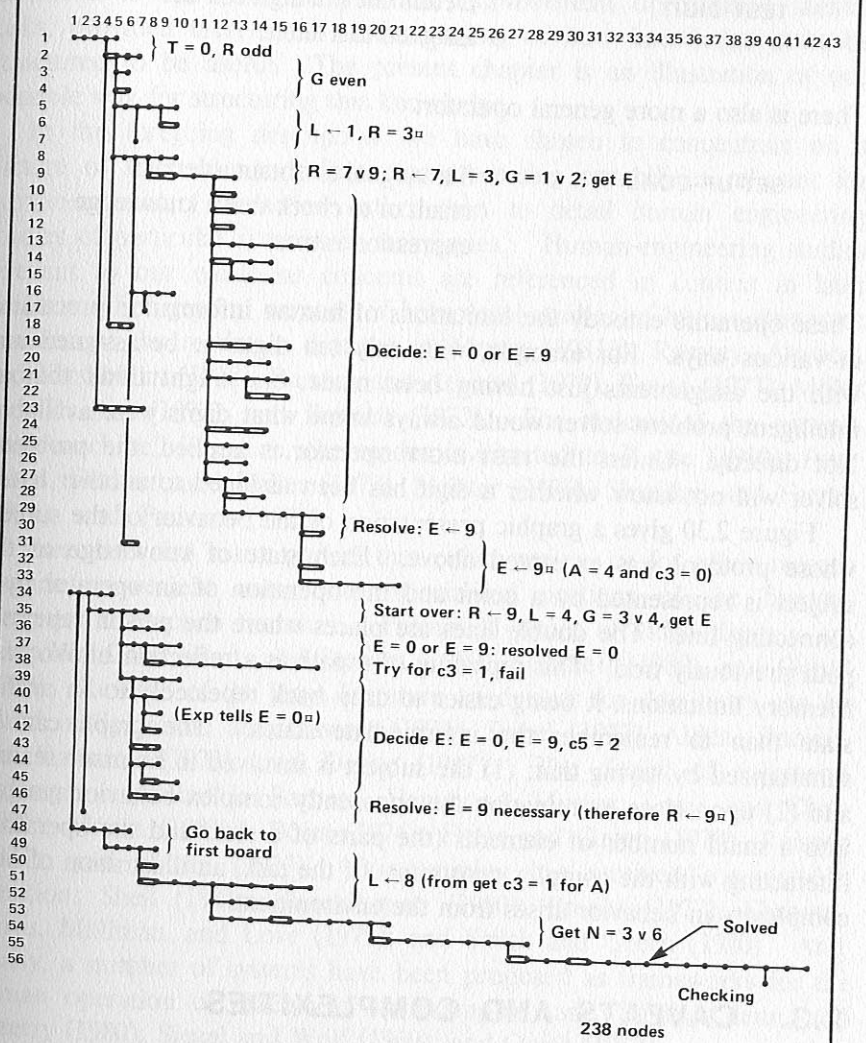

Figure 2.30. Search of subject through his internal problem space for the cryptarithmetic task. Subject S3, Newell and Simon (1972, Figure 6.4, P. 181) for DONALD + GERALD = ROBERT. Each dot in the diagram represents a state of knowledge of the subject. Each link is the result of applying an operator.

GENERATE-DIGITS	Determine what numbers are possible for a letter.
TEST-DIGIT	Determine if a digit can be assigned to a letter.

There is also a more general operator:

SET-UP-GOAL	Set up goal to obtain a certain result or to check that a knowledge expression is true.

These operators embody the limitations of human information-processing in various ways. For example, with only ten digits to be assigned and with the assignments just having been made, one might think that an intelligent problem solver would always know what digits were available. Not directly. Unless the **TEST-DIGIT** operator is applied, the problem-solver will not know whether a digit has been assigned to another letter.

Figure 2.30 gives a graphic presentation of the behavior of the subject whose protocol was excerpted above. Each state of knowledge of the subject is represented by a point and the operation of an operator by a connecting line. The double lines are places where the person repeats a path previously trod. This repeating of a path is a reflection of Working Memory limitations, it being easier to drop back repeatedly to an anchor state than to remember the intermediate states. The graph can be summarized by saying that: (1) the subject is involved in heuristic search; and (2) upon close examination the apparently complex behavior resolves into a small number of elements (the parts of a state and the operators) interacting with the complex constraints of the task, an illustration of how complexity in behavior arises from the environment.

2.3. CAVEATS AND COMPLEXITIES

We have attempted to convey a version of existing psychological knowledge in a form suitable for analyzing human-computer interaction. We have summarized this knowledge in a simple model of the human processor and have suggested, through examples, how it might be used with task analysis, calculation, and approximation to support engineering calculations of cognitive behavior. Although it is hoped that the model

itself will be useful, the real point is in the spirit of the enterprise: that knowledge in cognitive psychology and related sciences is sufficiently advanced to allow the analysis and improvement of common mental tasks, provided there is an understanding of how knowledge must be structured to be useful. The present chapter is an illustration of one possible way for structuring this knowledge.

In the foregoing description, we have chosen to concentrate on a picture of basic human information-processing capabilities relevant for human-computer interaction rather than to detail human engineering studies of particular systems or techniques. Human-engineering studies relevant to our particular concerns are referenced in context in later chapters. For general reviews of behavioral studies of human-computer interaction, the reader is directed to Moran (1981*b*), Ramsey, Atwood, and Kirshbaum (1978), Ramsey and Atwood (1979), Rouse (1977), Miller and Thomas (1977), and Bennett (1972). For reviews of the general "man-machine" literature, the reader is directed to Rouse (1980), Pew, Baron, Feehrer, and Miller (1977), Meister (1976), Sheridan and Ferrell (1974), and Parsons (1972).

There are also many papers that either review, or for other reasons provide convenient entry into, specialized portions of the human-computer interaction literature. Perceptual issues of video displays are treated in Cakir, Hart, and Stewart (1980), Shurtleff (1980), and Gould (1968). Reviews of the large literature on devices for data entry can be found in Sperandio and Bisseret (1974), Seibel (1972), Alden, Daniels, and Kanarick (1972), and Devoe (1967). The design of command languages is treated in Barnard, Hammond, Morton, Long, and Clark (1981); Moran (1981*a*); Boies (1974); Fitter and Green (1979); Reisner (1981); and Martin (1973). Programming has received considerable attention: Sheil (1981); Shneiderman (1980); Brooks (1977); Shepard, Curtis, Milliman, and Love (1979); and Smith and Green (1980). And finally, a number of systems have been proposed as frameworks for the human operation of machines; for example, Lane, Streib, Glenn, and Wherry (1980); Siegal and Wolf (1969); and Quick (1962).

The model of human information-processing that we have presented is our own synthesis of the current state of knowledge. In many respects (though not all) it corresponds to the dominant model of the seventies (Fitts and Posner, 1967; Neisser, 1967; Atkinson and Shiffrin, 1968; Welford, 1968; Newell and Simon, 1972; Lindsay and Norman, 1977; Anderson, 1980). But beyond any general model, a large amount of detailed knowledge is available in the literature on all the phenomena we have examined. In order to make the reader aware in some general way of the limits of our model, we mention briefly a number of the complexities documented in the literature and some of the alternative theoretical views.

BOXES VS. DEPTH OF PROCESSING

The dominant model of the seventies had as an underlying heuristic the assumption that there was an elaborate logic-level structure of many separate registers (the "boxes"), each with its own distinct memory parameters and connected by a distinct set of transfer paths. There was a Short-Term Memory consisting of seven chunks, brought into prominence by Miller (Miller, 1956; cf. Blankenship, 1938); forgetting was accomplished by displacement from fixed slots in the registers. Short-Term Memory was separate from Long-Term Memory, in contradistinction to the earlier theory, which simply posited a single structure of stimulus-response connections. The discovery by Sperling (1960) of the Visual Image Store, which was clearly distinct from the Short-Term Memory, provided impressive support for the "box" view.

A number of difficulties have beset this model, mostly in increased complexities and muddying-up of initially clean distinctions, as experimental evidence has accumulated. Initially it appeared that all information in the Short-Term Memory was coded acoustically (Conrad, 1964) and all information in Long-Term Memory coded semantically, but this has proved not to be the case. For instance, in some of the examples in this chapter, the use of visual codes in Working Memory is evident. Initially, rehearsal seemed to play the key role in the transfer of information from the Short-Term Memory to the Long-Term Memory—the more an item was rehearsed, the better chance it had of being stored away permanently. It has since seemed necessary to distinguish maintenance rehearsal, which has no implications for permanent memory, from elaborative rehearsal, which does. This distinction proved to be the crack in the edifice. It resulted in a new general view, called *depth of processing*, which attempts to do away with the structural boxes entirely and substitute a continuum of processing depth to determine how well material is remembered. "Depth" is defined somewhat intuitively: examining the letters of words is shallow, finding rhymes a little deeper,

and creating stories using the words deeper still. This view is now itsel under serious attack (Wicklegren, 1981) for lack of precision in its theor and for its unsuccessful predictions.

WORKING MEMORY SPAN

The original view of Working Memory, following Miller (1956), wa that it had a capacity of 7±2 items, coinciding with the immediat memory span. Gradually, much of the support for the existence of a independent Working Memory came from the recency effect in fre recall (the fading ability to remember the last few items heard that w examined in Figure 2.24). Various ways of calculating Working Memor size from the recency effect all give answers in the range 2.5~4.1 item for the capacity. This implies that the immediate memory is a compoun effect of more than one process, which is the way we have described it

At the opposite end of the spectrum from sizes of 2.5~4.1 vs. 7±2 i the notion of Working Memory as an activation of Long-Term Memory hence, of essentially unlimited instantaneous extent, but of limited acces The model presented here couples such a view with that of decay to ge the limited access. This view, though not widely stated explicitly, i represented in a few places in the literature (Shiffrin and Schneide 1977).

The Model Human Processor has moved some distance from th model of the early seventies in replacing separate memory registers wit registers that are subregisters of each other: Working Memory is th subset of activated nodes in Long-Term Memory, and the Visual an Auditory Image Stores are not completely separate from Workin Memory. Baddeley (1976, 1981) and his co-workers have used the term Working Memory functionally to include additional components of th human limited-capacity short-term storage system, which combine fo skilled tasks such as reading to provide a capacity somewhat larger tha our μ_{WM}. Chase and Ericsson (1982) have used the term Workin Memory to include rapid accessing mechanisms in Long-Term Memory what we have termed Effective Working Memory. They showed in a series of ingenious experiments that, through extensive practice, peopl can enormously increase their Effective Working Memory beyond ou $\mu_{WM}*$. The upshot of the Baddeley and Chase and Ericsson results is t emphasize the intimate connection between Working Memory, Long-

Term Memory, and attention. For the sake of simplicity, we have not attempted to incorporate these ideas into the Model Human Processor, pending their further development.

MEMORY STRENGTH VS. CHUNKS

The notion that memories have strengths, and can be made stronger by repetition, has been a central assumption of much psychological theorizing. Wicklegren (1977) gives a good account of this view for the whole of memory. The notion that memories come in discrete chunks, which either exist or do not exist in Long-Term Memory, provides an alternative conception that has risen to prominence with the information-processing view of man. It is this view we have presented.

It is difficult to determine in a simple, experimental way which of these two positions holds in general. Each type of theory can mimic and be mimicked by the other. One basic difficulty is that memory phenomena, being inherently errorful and varying, always lead to data samples that show considerable variation. One can never tell easily whether the variation arose from corresponding variation of strength or from discrete probabilistic events. The same effects producible by gradation in strengths also flow from multiple copies of chunks (Bernbach, 1970). Such multiplicity, far from being contrived, might be expected if a system manufactured chunks continually from whatever was being attended to.

WHAT IS LIMITING?

That humans are limited in their abilities to cope with tasks is clear beyond doubt. Where to locate the constraint is less clear. One general position has focused on memory as the limiting agent, as in the notion of the register containing a fixed set of slots. Another general position has focused on processing. A more sophisticated notion is that processing and memory may each be limiting but in different regions of performance (Norman and Bobrow, 1975). The processing position has usually taken the form of some sort of homogeneous quantity called *processing capacity*, which is allocated to different tasks or components of a task, usually within a *parallel* system. Another form of processing limit is to posit a *serial* system and permit it only one operation at a time.

Again, it is not possible to formulate experimental ways of distinguishing these alternatives in general. Serial processing systems can

mimic parallel ones by rapid switching, and parallel systems of limited capacity can show the most obvious sign of serial processing, linear time effects.

INTERFERENCE VS. DECAY

The Model Human Processor incorporates spontaneous decay over time and interference as mechanisms that produce memory-retrieval failure. Typically these are held to be alternative mechanisms and much effort has gone into trying to determine to which one forgetting is attributable. Actually, with the advent of information-processing models, a third alternative occurred: *displacement* of old items by new ones. This is clearly a version of interference, though one that involves total loss at storage time (of the interfered-with item), not of interaction at retrieval time.

The strong role of interference in long-term forgetting has been well-established. However, no one has ever accounted for the losses in very long term memory (weeks, months, or years) in a way that excludes genuine forgetting, although at least one investigator (Wickelgren, 1977) believes he can separate true forgetting from interference in the long term.

EXPANSIONS OF THE MODEL HUMAN PROCESSOR

There are at least three areas where the description of the Model Human Processor might be significantly expanded at some cost in simplicity. The first area is the semantic description of Long-Term Memory. As the study of Long-Term Memory proceeded, it became evident to psychologists that, in order to understand human performance, the semantic organization of Long-Term Memory would have to be taken into account. We have not described semantic memory in any depth here, since the details of such an account would carry us beyond the bounds set for this chapter. For surveys of the relevant literature, the reader is referred to Anderson (1980), Lindsay and Norman (1977), Norman and Rumelhart (1975), and Anderson and Bower (1973).

The second area is the description of the Perceptual Processor. In the simplified description we have given of perceptual processing, we have skipped over considerable detail that is appropriate at a more refined level of analysis. A description based on Fourier analysis could be used to replace various parts of the model for describing the interactions of visual stimuli with intensity and distance (Cornsweet, 1970; Ganz, 1975; Breitmeyer and Ganz, 1976).

The third area is the description of the Cognitive Processor. We have not said much in detail about the control structure of the Cognitive Processor; but it is necessary to consider the processors's control discipline if interruptability, errors, multiple-tasking, automaticity, and other phenomena are to be thoroughly understood. A more detailed description of the recognize-act cycle, and how the characteristics of simple decisions arise from it, might be given in terms of a set of recognize-act rules, called *productions* (Newell, 1973). According to this description, the productions themselves reside in Long-Term Memory. On each cycle, the recognition conditions of the rules are compared with the contents of Working Memory (or said another way, some of the recognition conditions of the rules are activated through spreading activation in Long-Term Memory). The rule with the best match (the highest state of activation) fires and causes its associated action to occur, altering the contents of Working Memory (activating other chunks in Long-Term Memory). Perceptual input whose recognition activates previously non-activated chunks in memory may, through this mechanism, interrupt and redirect the previous course of processing. The description might be elaborated to give both an account of skilled behavior that requires little conscious attention and an account of unskilled behavior. A production system description has also been used to give a description of complex information-processing where each action might involve several dozen recognize-act cycles (for examples, see Newell and Simon, 1972; Young, 1976; Anderson 1976).

THE EXISTENCE OF ALTERNATIVES

Does the existence of alternatives to various features of the Model Human Processor, like those we have just mentioned, and the fact that agreement on them is very difficult to obtain, rob the model of its usefulness or show that it is impossible to settle things in psychology? Not at all, and for two reasons.

The first reason is a technical issue about making progress in psychology. Many of the difficulties arise because classes of quite different mechanisms can mimic each other rather closely, as in the case of interference and decay. However, this mimicking works only over narrow ranges of behavior. For instance, if only one specific task is considered—say, the immediate memory distractor task (Figure 2.6) in

which a single item is given, then counting backward by sevens, the attempting to recall the item—it is easy to generate several explanatio (decay, interference, displacement) that are indistinguishable, even principle, by unlimited precision in the data. But if these san mechanisms are required to provide the explanation in many divers tasks, it becomes much harder for the mimicking to succeed. Thus, th comments we have made apply locally—mechanism X competes wit mechanism Y to explain a given phenomenon, but only when tha phenomenon is considered in relative isolation.

The current style in psychology is to have a highly elaborated base c quantitative data over many diverse phenomena, with many loca theories. The science has not yet succeeded in putting together genera theories that are tight enough quantitatively so that the same posite mechanism (for example, Working Memory decay) is forced to sho itself in action in a large diversity of tasks. Such comprehensive theorie may soon emerge—the groundwork seems well-laid for them—but ther has not yet been enough of this theorizing to settle the issues reflected i this section.

The second reason that the existence of alternatives does not rob th model of its usefulness concerns the use to which our model is to be pu The model's purpose is to provide a sufficiently good approximation t be useful. Its function is synthesis, not discrimination of alternativ underlying mechanisms. If basic mechanisms are not distinguishable in domain where there has been extensive empirical investigation, there i some assurance that working with either will provide a reasonable firs approximation. Then it is important to obtain a single overall pictur based on one set of mechanisms that works globally and fits in with a appropriate unified theoretical perspective. This we have done.

Our purpose in this chapter has been to prepare the way for th specific set of studies of human-computer interaction that is to follow Though these studies do not take the details we have been presenting fo granted, they do presume the basic orientation laid out here.