HUMAN–COMPUTER INTERACTION, 1996, Volume 11, pp. 321–355

TYPIST: A Theory of Performance in Skilled Typing

Bonnie E. John
Carnegie Mellon University

ABSTRACT

TYPIST is a TheorY of Performance In Skilled Typing built within the framework of the Model Human Processor (MHP; Card, Moran, & Newell, 1983). As such, it can be used to make quantitative predictions of performance on typing tasks and can be integrated with other MHP-based models of performance. In this article, I present the theory and explain the source of each theoretical assumption (MHP, typing task analysis, or empirical typing data). I then demonstrate different ways to use TYPIST by applying it to six transcription typing tasks. Finally, I summarize its application to many more typing tasks that display robust behavioral phenomena identified by Salthouse (1986).

1. INTRODUCTION

Despite advances in direct manipulation, speech and gesture recognition, and other input techniques, typing is still by far the most prevalent form of human-to-computer communication today. Therefore, it is important to understand that skill sufficiently to allow consideration of human typing capabilities in the design of many computer systems.

Typing has been studied extensively by psychologists for almost a century, and scores of laboratory experiments have been done under as many different experimental conditions. In 1986, Timothy Salthouse re-

Bonnie E. John is an engineer and psychologist with an interest in predictive models of human performance; she is an Assistant Professor in the Human–Computer Interaction Institute of Carnegie Mellon University.

CONTENTS

viewed this vast literature and proposed a set of 29 robust phenomena (listed in Section 3.7) that a complete theory of typing would have to explain. Since then, dozens more studies have been done, confirming, refining or providing more details about the phenomena. Salthouse's article, however, stands as the most comprehensive review of this skill.

Similarly, there are many psychological theories of typing: conceptual and computational; mathematical, symbolic, and connectionist. For example, Salthouse (1986) proposed a model of transcription typing able to provide qualitative explanations for the phenomena he identified.[1] Rumelhart and Norman's (1982) connectionist model of typing is embodied in a computer simulation and provides detailed predictions about the movement of fingers, the relative response times for letters in different contexts, and several types of errors. Sternberg, Knoll, and Wright's (1978) subprogram-retrieval model accounts for several phenomena associated with discontinuous typing including the dependence of latency and interkeystroke interval on string length, and a serial position effect on interkeystroke interval. Pashler (1994b) delves into the implications of a serial bottleneck in the response selection aspects of continuous typing, especially with respect to restricted preview and dual-task performance. Heath and Willcox (1990) demonstrated a stochastic process model of interkeypress time that may be useful in identifying fatigue or occupational overuse. All of these models have explained some fraction of empirical

[1] Salthouse (1986) stated that his model was "derived from ideas introduced by earlier theorists (e.g., Cooper, 1983; Logan, 1983; Rumelhart & Norman, 1982; Shaffer, 1973, 1975, 1976; Shaffer & Hardwick, 1970; Thomas & Jones, 1970)" (p. 303).

typing data and have engendered more research and theoretical discussions of mechanism, such as discussion of the role of central and peripheral timing mechanisms (Gentner, 1987) and the discussion of overlapping mental operators following Pashler's article (McLeod & Hume, 1994; Pashler, 1994a).

However, for a theory of typing to be useful in the design of computer systems, it needs to do more than explain laboratory-generated typing data or spark scientific discussion. Because computer users type in service of larger tasks, such an engineering model needs to be integrated with models of other activities to model complex tasks. Because design information is needed before a system is operational, an engineering model of typing needs to make quantitative predictions of performance in the absence of empirical data on the actual task. It must be straightforward enough to be usable for system designers who are not experts in psychology. Finally, it should be useful at different levels of approximation (i.e., analyst's effort) depending on the requirements of the design situation. None of the models just summarized were ever intended to be engineering models and thus do not meet all these criteria in one way or another.

To these ends, I present TYPIST,[2] an engineering model of expert-transcription typing that covers many of the phenomena identified by Salthouse. TYPIST is derived from the Model Human Processor (MHP; Card et al., 1983) and a task analysis of transcription typing. Given the expected typing speed of a user, TYPIST can make quantitative, a priori predictions of performance on many typing tasks. It can be integrated with other perceptual and information-processing activities that the MHP can address. It also can be used at different levels of approximation, from the well-known Keystroke-Level Model (KLM; Card, Moran, & Newell, 1980) down to individual finger movements described by Fitts's Law. TYPIST has also served as the basis or inspiration for several models of more complex real-word tasks that involve parallel cognitive, perceptual, and motor activities (e.g., Gray, John, & Atwood, 1993; Nelson, Lehman, & John, 1994).

Section 2 reviews the MHP (the foundation of TYPIST) and presents the typing-specific assumptions that are the heart of TYPIST. Section 3 applies TYPIST to six typing tasks to demonstrate different uses of the theory. Finally, Section 4 summarizes TYPIST's account of Salthouse's robust phenomena and point to how it can be extended and used in real-world design.

[2] Many thanks to David Rosenbaum for suggesting the acronym. This model was previously called a Model of Expert Transcription Typing (METT; John, 1988, 1993), a much less mnemonic name.

2. MODEL HUMAN PROCESSOR (MHP) AND TYPIST

TYPIST is comprised of a set of assumptions about how the MHP's processors work together to perform transcription typing. These assumptions are derived primarily from the dictates of the typing task and the workings of the MHP rather than from empirical typing data. Only one assumption was derived with reference to qualitative typing data and no other theories of typing were used to guide the assumptions. Thus, TYPIST is a theory-driven model, rather than an empirically driven model, with the MHP being its direct intellectual predecessor.

In Section 2.1, I review the MHP and then present the assumptions of TYPIST, each with an explanation of how that assumption was derived. Just as the MHP is a gross simplification of the human perçeptual/cognitive/motor system–to enable quick but fairly accurate calculations of performance–TYPIST's assumptions are often gross simplifications of what we know to be true of this complex perceptual/cognitive/motor skill. The purpose of TYPIST is to provide predictions of typing behavior with a minimum of psychological mechanism so that it would be useful in the analysis of human–computer interaction (HCI) tasks; it is not intended to provide a definitive accounting of complex details. Thus, these assumptions favor conceptual simplicity; the more blatant of these simplifications are discussed in Section 2.2. After presenting the assumptions in prose, I present a graphical representation of TYPIST typing an example sentence that illustrates how these assumptions work together to model performance.

2.1. MHP: The Foundation of TYPIST

The MHP was presented by Card et al. (1983) in *The Psychology of Human–Computer Interaction* as an attempt to create a unified, integrated model of human behavior that would allow quantitative predictions of behavior, especially in the domain of HCI. In contrast to many psychological models, this engineering model could make absolute predictions using what was then known about human capabilities, without reference to empirical measurements on the task in question. Card et al. presented 18 examples of using the MHP to make such predictions, from computing the refresh rate necessary to produce the illusion of movement in computer animation, to the efficient placement of keys on a handheld calculator, to remembering arbitrary file names.

The MHP is specified in terms of three processors (perceptual, cognitive and motor), four memories (two very short-term perceptual stores, working memory, and long-term memory) and a few quantitative parameters of each (e.g., the cycle time of the processors and the capacity of the memories). The processors work serially within themselves but in parallel with each other, subject to serial limitations imposed by data flow requirements (e.g., the cognitive processor may need information from the perceptual processor before it can proceed with a task).

The processors and memories of the MHP work together under the control of 10 principles of operation. These principles cover many aspects of the operations of the three processors, ranging from statements about the cycle times of the processors to a general statement that problem solving takes place in a problem space. They are derived from psychological evidence (e.g., a preponderance of data confirming Fitts's Law) and from a specific model of the psychological world (e.g., the rationality principle). TYPIST uses only a few of these principles directly, as discussed in Section 2.2.

TYPIST makes use of these constructs, and parameters set in previous research, to produce predictions of behavior in the typing tasks. In particular, work by John and Newell (1990) in the domain of stimulus–response compatibility was specifically intended to produce estimates of operator durations that would be transferable to other tasks. Such estimates allow a priori, quantitative predictions of performance to be made about tasks that use those operators. Three of the operators estimated by John and Newell are relevant to the task of transcription typing (Figure 1). The first is a reestimate of the MHP's cognitive cycle time using experiments specifically designed to reveal a value for a minimal cognitive operator. The remaining two are composed of several actions of the MHP processors and are operationally defined:

1. A perceptual operator perceives a written word and encodes it into an ordered list of letters that is the spelling.
2. A motor operator finds and hits a key on a keyboard–assuming an unskilled typist of about 30 gross words per minute (gwpm).[3]

These operators and their duration estimates will be transferred directly to TYPIST and used to make quantitative predictions of performance.

2.2. Assumptions of TYPIST

For an engineering model of typing to be useful in the design of computer systems, it need not be distinctly different from all other typing models or make novel predictions. In the presentation of TYPIST that follows, many aspects of other typing models will be evident. For example, the perceptual, cognitive and motor processes of the MHP also appear in

[3] The measure of typing speed called *gross words per minute* (gwpm) is uncorrected for errors.

Figure 1. **Perceptual, cognitive, and motor parameter definitions and estimated durations.**

Parameter	Definition	Duration
Perceptual operator	Reading a word of about six letters and encoding it into an ordered list of letters	340 msec
Cognitive operator	A cognitive processor cycle time	50 msec
Motor operator	Typing a character on an alphanumeric keyboard at a rate of about 30 gwpm	230 msec

the work of several typing theorists (e.g., Pashler, 1994b; Salthouse, 1986; Shaffer, 1975). Also, there is a serial bottleneck in the MHP, and therefore in TYPIST, as discussed in Pashler (1994a, 1994b) and McLeod and Hume (1994). Just as the MHP combined many theories of human performance into a simple but unified theory for answering questions in system design, TYPIST's emphasis is on breadth and usefulness rather than novelty. TYPIST did not combine these theories directly; rather, the similarity came from the constraints placed on TYPIST by the MHP and the typing task itself.

> ***Assumption 1: Basic method.*** The basic method is that TYPIST perceives a *chunk* from the physical manuscript to be typed (a word, syllable, or letter; see Assumption 4) and encodes it into an ordered list of characters (the spelling) with a perceptual operator. If it is a word or syllable, a cognitive operator retrieves the spelling of that chunk from long-term memory (LTM). The first character in the list is initiated with a cognitive operator and then executed with a motor operator. The second character is then initiated and executed. This process continues until the chunk is complete. If a letter is perceived alone, then the letter is initiated immediately following the perception and executed.

This basic method follows from a task analysis of transcription typing and the operation of the MHP. In the basic transcription typing task, the typist starts with a physical manuscript and ends with physical keystrokes. Working backwards, the only way the MHP's motor processor can type a keystroke is if there is a symbol in working memory (WM) telling it to do so. The MHP's cognitive processor can deposit such symbols, but it can only manipulate symbols already in WM or LTM. Thus, the symbol for what to type from this manuscript must be deposited in WM by the MHP's perceptual processor. If that symbol is a single letter, no other cognitive operations are necessary besides tagging it as the next desired keystroke. If the symbol is more complex, such as a word or a syllable, then the cognitive processor must retrieve its component letters from LTM before it can put the single characters into WM for the motor processor to act upon.

> ***Assumption 2: Serial/parallel processing.*** TYPIST starts with the MHP's restriction that each of the processors work serially within itself. It adds the following typing-specific assumptions.
>
> ***Assumption 2a. Perception/cognition interaction.*** Perception has to be complete before getting the spelling or initiation of a character can begin.

This assumption comes from the MHP's serial processors and its first principle of operation, the recognize–act cycle. The MHP assumes a cycle of action where the contents of WM are acted upon, changed, and then acted upon again. Thus, the perceptual processor must complete its cycle and deposit a symbol in WM before the cognitive processor can act on it. There are several other modeling paradigms that do not assume such a discrete cycle–such as McClelland's (1979) cascade model and Rumelhart and McClelland's (1986) parallel distributed processing (PDP) models, but this simplifying assumption has been shown to be adequate to predict interesting performance characteristics of human behavior in many tasks (Card et al., 1983).

> ***Assumption 2b. Same-hand constraint.*** A character on the same hand cannot be initiated with a cognitive operator until the motor processor execution of the previous character is complete.

This is the only assumption in TYPIST derived primarily from empirical data on typing itself. Salthouse's (1986) Phenomenon 11 states that alternate-hand keystrokes are faster than same-hand keystrokes; this qualitative effect is achieved by Assumption 2b. Quantitative data behind this qualitative statement shows an almost constant difference between same- and different-hand interkey time of about 40 msec across a wide range of typing skill (Ostry, 1983). The quantitative effect of Assumption 2b is to insert a 50 msec cognitive operator between same-hand keystrokes but not between alternate-hand keystrokes, thereby approximately matching the empirical results. TYPIST accepts this approximation in the interest of simplicity.

Assumption 2b and the underlying MHP assumption (that the motor processor operates serially) produce a model that moves one finger at a time, pausing briefly between same-hand finger movements. Anyone who has ever watched a skilled typist knows that this model is wrong, that many fingers are moving at once. However, for many typing tasks, the overlap between fingers is not a primary contributor to performance and a discrete model like TYPIST is adequate to predict important performance measures. In Section 3.6, I explore the use of TYPIST on a typing task that requires the modeling of individual fingers and how they overlap. When the task demands, Assumption 2b can be relaxed and Fitts's Law can be added to TYPIST to make finer predictions of same-hand interkey time.

> ***Assumption 2c. Perception/WM limitation interaction.*** The perceptual processor does not perceive the next piece of information unless there is room in WM for that information.

The MHP's WM has a limited capacity (see Assumption 3). This assumption is derived in response to that limitation to ensure that TYPIST will not exceed this capacity by allowing perception to get too far ahead of the motor processor.

> ***Assumption 3: WM limitation.*** In normal transcription typing, TYPIST assumes that the perceptual processor stays three chunks ahead of the cognitive processor. The chunk is usually a word, but it can be a syllable or a character if words are not available in the specific typing task.

This three-chunk limitation is the pure capacity of WM proposed in the MHP. These chunks may access LTM for the spelling of one chunk at a time, extending the effective capacity of WM to, on average, seven chunks; assuming an average word length of five letters, the seven chunks would be the five letters of the first word, plus the two next words.

> ***Assumption 4: Perceived chunks.*** The perceptual processor deposits a chunk in WM at the most meaningful level available at or below the word level. For example, if words are present, they are perceived and encoded and a word-chunk is put in WM. If the view of whole words is restricted or if there are no words present (as when typing random letters), the perceptual processor produces syllable-level chunks. If syllables are not visible because of restricted view or random characters, then the perceptual processor produces single characters. This assumption follows from the discussion of chunks and WM in Card et al. (1983, pp. 36–37).

> ***Assumption 5: Cognition/motor interaction.*** Once a character is initiated with a cognitive operator, the motor operator that executes that character cannot be stopped. Again, this assumption follows directly from the MHP (see the discussion of the motor system in Card et al., 1983, pp. 34–35).

Assumption 6: Operator similarity. Across all domains to which the MHP has been applied, similar operations involving similar perceptual, cognitive, and motor operators take similar amounts of time. This assumption is fundamental to the MHP, with its characteristic processor cycle times.[4] Thus cognitive and perceptual operators and their duration estimates are transferred directly from the stimulus–response compatibility work of John and Newell (1990), as shown in Figure 1, to TYPIST in the next two assumptions. (The motor operator duration in Figure 1, derived from inexperienced typists, serves as an extreme upper bound for expert-typist motor operators, and is used in the next section to estimate more realistic expert-typist motor operator durations.)

Assumption 6a. Perceptual operator duration. The time to perform a perceptual operator (perceive a visual stimulus) is 340 msec. A simplifying assumption is that this time is constant whether the thing to be perceived is a word, a syllable, or a character.

Assumption 6b. Cognitive operator duration. The time to perform a cognitive operator is 50 msec.

Assumption 7: Motor operator duration and interaction with skill. As a simplifying assumption, practice in typing decreases the motor operator time only; the estimates of the perceptual and cognitive operators remain constant. Given the amount of practice an adult typist has had perceiving words (as a part of reading) and in the cognitive operations involved in spelling words (as a part of writing), the power law of practice (MHP's principle of operation P6) suggests that these operators remain relatively constant as compared to the more newly acquired motor operators of typing.[5]

[4] The MHP (Card et al., 1983) does include room for some variation of times. For example, principles of operation P1 and P4 allow variation in the perceptual and cogntivie processor cycle times, respectively, due to differences in task situations, and individual differences can be explored in a limited way by using the Fastman/Middleman/Slowman estimates. However, TYPIST does not use the more detailed principles and has not made use of Fastman/Middleman/Slowman.

[5] An alternative assumption is that all the operator durations stay constant, but overlap between operators increases with typing skill (Salthouse, 1984a). Detailed analysis of videotaped skilled typists suggest that overlap between fingers almost certainly increases with increased typing speed, but increases in single-finger tapping rate and typing speed of individually presented letters suggest that decrease in the motor operator also plays a role. Given the other assumptions, TYPIST already overlaps the perceptual, cognitive, and motor processors as much as it can, and overlap between fingers is precluded by Assumption 2b (but see Section 3.6 for further discussion of this point). Thus, this explanation lies outside the scope of TYPIST. However, the existence of alternative models does not "rob a model of its usefulness" (Card et al., 1983, pp. 96–97).

2.3. Estimating the Motor Operator

As discussed in Section 2.2, TYPIST assumes that the duration of the motor operator decreases with typing skill. To make predictions of different typing tasks with differently skilled typists, we need to provide estimates for the motor operator associated with different typing speeds. This derivation of the motor operator also serves as an introductory example of how to use TYPIST.

Consider a typist typing a sentence from a standard typing test,[6] the *example sentence,* which is used in analyses throughout this article: `One reason is quite obvious; you can get in and get out without waiting for the elevator.`

The first three words are perceived with three perceptual operators; the spelling of the first word is retrieved from LTM with a cognitive operator; and the letters of the word, and the space following it, are initiated and executed in turn. As soon as the space has been initiated, the chunk is out of WM, making room for the next word, so the perceptual processor perceives the next word. The processes continue until the entire sentence is typed.

The parallel operation and sequential dependencies of the three processors make the processes of typing difficult to analyze and talk about. Fortunately, there is an analysis technique borrowed from engineering project management that allows easy analysis of parallel resources (the three processors) working with sequential dependencies (outlined by the typing-specific assumptions). The technique is called the critical path method (Stires & Murphy, 1962).[7] The particular version of this technique used for the analyses in this article is embodied in a project management software package for the Apple Macintosh family of computers, MacProject.[8]

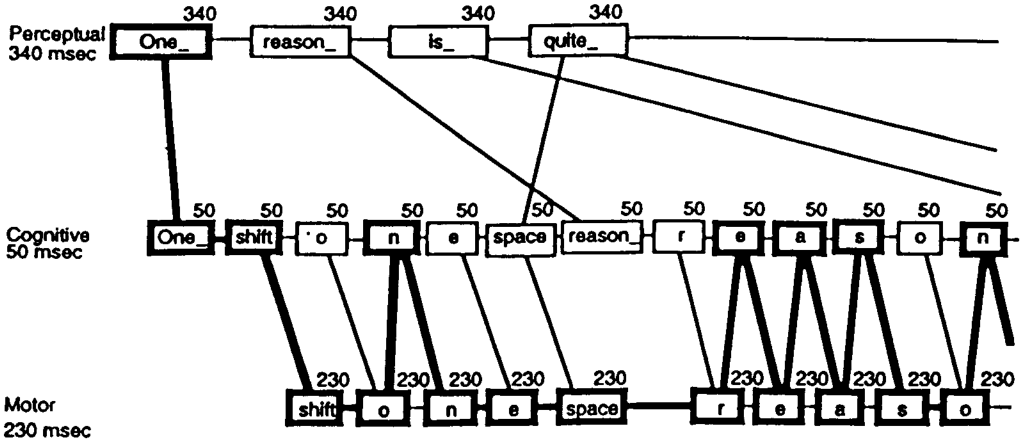
In a critical path analysis, each *subtask,* or *operator,* necessary to accomplish a total task is represented as a box with a duration. For instance, in Figure 2, the box labeled "One_" in the Perceptual row represents the perceptual operator that gets the word from the copy to be typed and has a duration of 340 msec. Dependencies between subtasks are represented by lines connecting the boxes. In Figure 2, the cognitive operator that retrieves the spelling of the word "One" (represented by the box labeled "One_" in the Cognitive row) can not be started until the word is perceived, so a line is drawn from the box labeled "One_" (Perceptual) to the box labeled "One_" (Cognitive). The representation of a full task is called

[6] The typing test was obtained from the Carnegie Mellon Univeristy Personnel Office, 1987.

[7] The critical path method has been used in other psychological models to represent parallel activities (e.g., Schweickert, 1978, 1980).

[8] Apple, Macintosh, and MacProject are trademarks of Apple Computer, Inc.

Figure 2. **A schedule chart and critical path for the example sentence for a 60-gwpm typist with an initial motor operator estimate of 230 msec.**



a *schedule chart.* The critical path is that set of operators that determine the duration of the total task (bold).

I use critical path analysis to make estimates of the motor operator for typists of different speeds. By definition, a 60-gwpm typist would be able to type the 89 characters of the example sentence in 17,800 msec. The schedule chart for typing the example sentence (Figure 2) is drawn up with all perceptual operators along the top row, with the boxes labeled with the word the typist sees in the copy to be typed (i.e., the example sentence). The cognitive operators are in the center row, labeled with the result of retrieving the spelling of a perceived word (in which the content of the box is the word previously perceived) or the character to be initiated. The motor operators are in the bottom row labeled with the character they are typing. The duration of the operators is set to 340 msec for perceptual, 50 msec for cognitive, and an initial guess of 230 msec for motor (the motor operator found in the stimulus–response (S–R) compatibility research from typists averaging 30 gwpm). The total duration of typing the sentence is calculated by the project management software to be 22,940 msec because there are 1 perceptual operator, 38 cognitive operators, and 90 motor operators on the critical path. This is too long for a 60-gwpm typist so, as expected, the initial guess for the motor operator duration is too high. Reducing the value of the motor operator and iterating through this process yields a schedule chart with the same critical path, and a motor operator estimate of 170 msec for a 60-gwpm typist.

An analyst gets qualitative information about the roles of the three processors by inspecting the schedule chart as well as numerical estimates from the critical path analyses. For instance, for a 60-gwpm typist, the perceptual operators are never on the critical path once the initial words have been perceived. This implies that the perceptual processes are not the limiting factors in the typing task. On the other hand, all the motor

Figure 3. A schedule chart and critical path for the example sentence for a 160-gwpm typist.

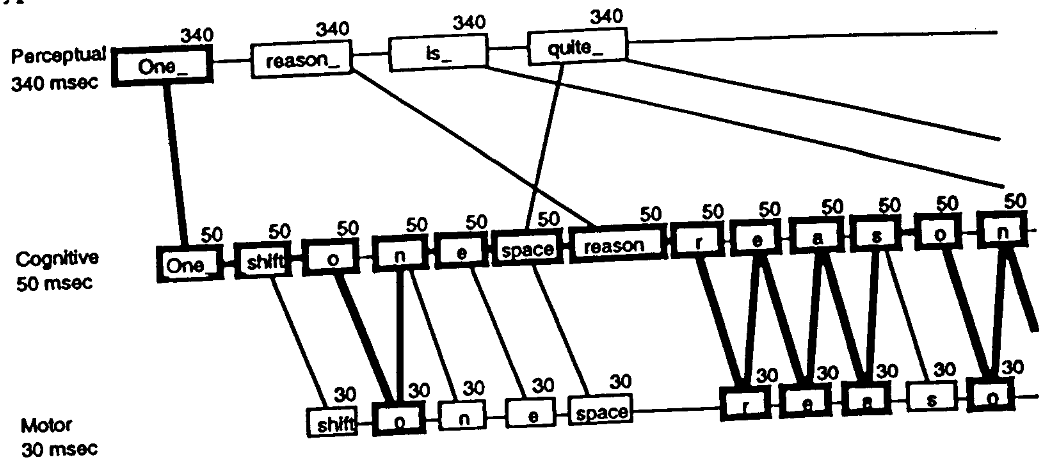
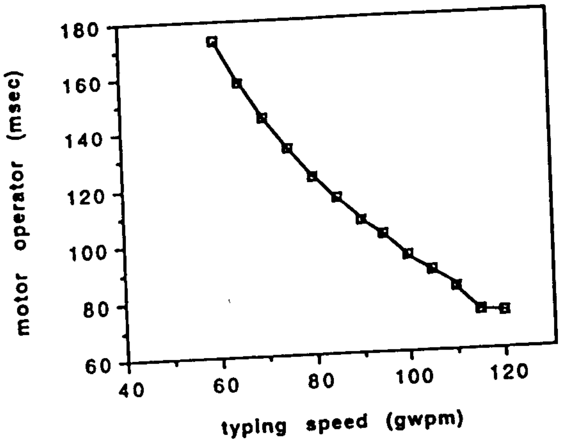


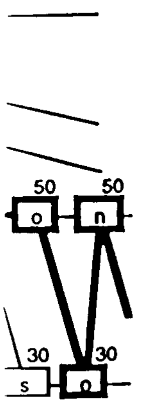
Figure 4. Typing speed versus motor operator estimate.



operators are on the critical path, indicating that a speed up of the motor operator will greatly affect the total time. At the other end of the speed spectrum–160 gwpm–the critical path (Figure 3) looks quite different, with the cognitive operators determining the critical path and the motor operators playing a much less important role. This is because the duration of the motor operator is now shorter than that of the cognitive operator. This implies a theoretical maximum for typing, 180 gwpm, at which the motor operator goes to zero, given the simplifying assumption that all the speedup with skill comes from a decrease in the motor operator (Assumption 6c). Indeed, most skilled typists do not even approach this speed, and the world record is only 20% above it (*Guinness Book of Records,* 1995).

Repeated application of this analysis process yields a chart (Figure 4) of estimates of the motor operator versus the gross speed of the expert typists used in the studies reviewed by Salthouse (1986). The motor operator estimates are rounded to the tens digit when used to analyze typing tasks.

160-gwpm



the motor
the speed
erent, with
otor opera-
ation of the
rator. This
the motor
hat all the
r (Assump-
speed, and
1995).
Figure 4) of
pert typists
or operator
ping tasks.

3. APPLYING TYPIST TO TYPING TASKS

Given the basic architecture of the MHP, the typing-specific assumptions of TYPIST, and the estimates of the motor operator durations, an analyst can take a particular typing task and build a quantitative model of performance on that task. Although TYPIST can be used for any number of typing tasks, I concentrate on those tasks that make particular points about how to use TYPIST.

- Reasoning about typing behavior at a high level is shown with an analysis of a *copying span* task.
- Using a TYPIST schedule chart and critical path calculation with all their details is demonstrated on an analysis of a *restricted preview* task.
- The use of a different view of the critical path, the timeline view, and how TYPIST predictions can be sensitive to small changes in the task is demonstrated on a *stopping span* task.
- A dual-task situation demonstrates how TYPIST can be integrated with other MHP models.
- A treatment of individual differences is discussed in a *detection span* task.
- Finally, I demonstrate how an analyst can modify TYPIST if the design situation requires a different level of analysis. Specifically, I show how TYPIST reduces to the KLM if differentiation between hands is not required, and how Assumption 2b (that fingers can only move one at a time) can be relaxed and Fitts's Law can be used with TYPIST to model overlapping fingers.

In summary, I briefly report the results of using TYPIST on 29 typing tasks identified by Salthouse (1986) as displaying robust performance phenomena (see Section 3.7).

3.1. High-Level Reasoning About a Typing Task

The copying span is defined as the amount of material that can be typed after a single inspection of the copy, and has been measured to be between two and eight words, or 7 to 40 characters. It has been measured in many different ways, and the different methods yield vastly different results, accounting for the wide range. For instance, Rothkopf (1980) measured the copy span by asking the typists to glance at the copy, remembering as much as possible, and then type it before glancing at the copy again. This is a very different task than normal transcription typing and it yielded the result that a typist can remember up to 40 characters at a time. Salthouse (1985) measured the copying span in a way more appropriate to transcription typing, and got an average copying span of 14.6 characters for expert typists (above 60 gwpm).

The Salthouse (1985) experimental situation was as follows:

> The procedure involved presenting material on the video monitor using the leftward-moving display with a preview window fixed at 39 characters. After a predetermined number of keystrokes, the display was erased and the typist instructed to continue typing as much material as he or she was confident appeared on the display. The material consisted of eight sentences, movie descriptions from *TV Guide* magazine, with an average length of 75 characters. Two sentences each were typed with 15, 25, 35, and 45 keystrokes prior to the disappearance of the display. The median number of characters that were typed correctly after the blanking of the display served as the measure of copying span. (Salthouse, 1985, p. 267)

TYPIST can model this task at several different levels of detail (see John 1988, 1993). Here, I present the simplest analysis to demonstrate how TYPIST can be used as a high-level engineering tool.

On average, a word is five characters long (that is, four actual letters and the space or punctuation after it). If there is a 3-word look-ahead, there are, on average, 15 characters in the perceptual buffer (i.e., WM). If the display is removed randomly, it will be removed, on average, 2.5 characters into a word. Therefore, there will be a copying span of about 2.5 words or 12.5 characters. This "quick and dirty" prediction is about 14% below the observed 14.6 characters.

This extremely simple analysis gives an adequate prediction of the copying span. The copying span task happens to depend so heavily on the WM assumption in TYPIST that no more than a "back-of-the-envelope" calculation is needed. However, this simple analysis is not detailed enough to predict most typing phenomena. A more detailed analysis procedure, which I demonstrate next, can be used to make more precise and subtle predictions.

3.2. Detailed Predictions With TYPIST

A particularly robust typing phenomenon is that typing rate is slower with restricted preview. That is, if the typist is prevented from looking ahead, his or her speed goes down. This phenomenon has been reported by many researchers (Hershman & Hillix, 1965; Salthouse, 1984a, 1984b, 1985; Salthouse & Saults, 1987; Shaffer, 1973; Shaffer & French, 1971; Shaffer & Hardwick, 1970). A typical task description is found in Salthouse (1984a):

> Task 3 was to type material displayed in a single line of the video monitor and arranged such that each keystroke caused the display to move one space to the left. No visible copy was produced in this task. In successive conditions, the display contained 19, 11, 9, 7, 5, 3, or 1 character of a 60- to

> 83-character sentence. The sentences were movie descriptions taken from *TV Guide* magazine and were randomly assigned to preview conditions. (Salthouse, 1984a, p. 350)

TYPIST's WM limitation and perceptual chunks assumptions are particularly relevant for this task. As the preview is restricted, the three-word look-ahead is cut back to two- and then one-word look-ahead, then down to the syllable level, and finally to the letter level. With a one-character preview, the task is reduced to a series of choice–reaction time (RT) tasks with no look ahead.

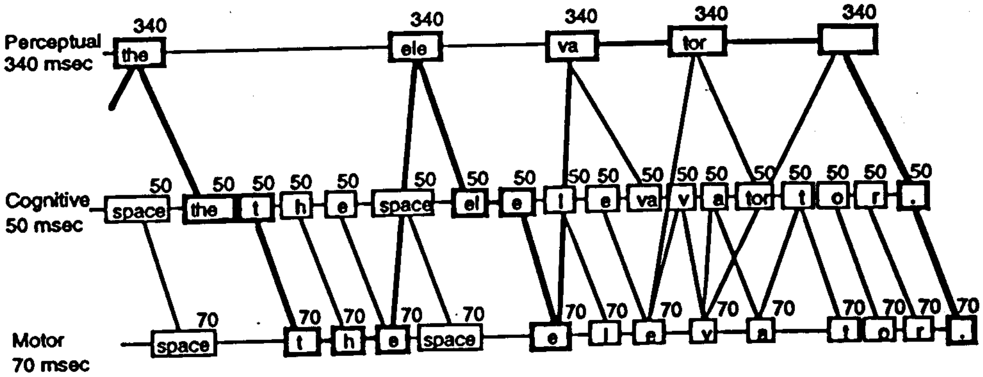
If a whole word does not fit in the preview window, then the perceptual operator will not start until the word is completed (or is shown not to fit in the window). For instance, if the window is nine characters long, then the first view of the example sentence would be `One_reaso` (Figure 5). In this case, "One" is perceived and encoded, but "reaso" is not. TYPIST waits until two characters are typed and the view becomes `e_reason_`; "reason" is now a whole word and a perceptual operator can work on it. Note that the perceptual operator does not start until the punctuation after the word is visible (in this case, a space) because the "reason" in `ne_reason`, for example, could easily be part of "reasonable" or another longer word.

If a word is too long to fit into the window completely, then the syllables in the window are perceived and encoded. The remaining syllables in the word are perceived and encoded as they appear until the word is complete. Separating the words into syllables for this analysis could be done from two different viewpoints: the user's viewpoint or that of the text. For example, with a five-character window, the phrase "the elevator" does not fit; after typing the "the", the view is `_elev` (Figure 5). The typist would not know what word was coming and might choose to encode this as either "e" and "lev", "el" and "ev", or "el" and "e" waiting for more letters before encoding the "v". On the other hand, the analyst knows text is actually "elevator" and this is commonly broken up as "el", "e", "va", "tor" (*Webster's New Collegiate Dictionary*, 1977). To ensure that TYPIST analyses are simple, straightforward, and objective, they are made from the viewpoint of the text as an approximation to what a user might actually do. Thus, `_elev` would be encoded as two syllables: "el" and "e". With a 3-syllable look-ahead, this sequence would not fill up WM. Therefore, as soon as the space is typed, the view would be `eleva` and the perceptual processor would perceive and encode the next syllable, "va". After the "l" is initiated, there would be room in WM, but the view would be `levat` and no new syllable would be visible. After the next "e" is typed, however, `vator` becomes visible and the "tor" is perceived and encoded. Then, after the "v", the view becomes `ator.` and the "." is perceived and encoded separately. Figure 6 shows the schedule chart depicting these dependencies.

Figure 5. The characters visible as a typist types the example sentence with different preview windows.

	Characters visible for different window sizes	
Character typed	9-character	5-character
Beginning	One_reaso	One_r
o	ne_reason	ne_re
n	e_reason_	e_rea
e	_reason_i	_reas
Intervening characters typed		
is quite obvious: you can get in and out without waiting for the		
e	_elevator	_elev
_	elevator.	eleva
e	levator.	levat
l	evator.	evato
e	vator.	vator
v	ator.	ator.

Figure 6. Schedule chart for "the elevator" in a five-character preview.

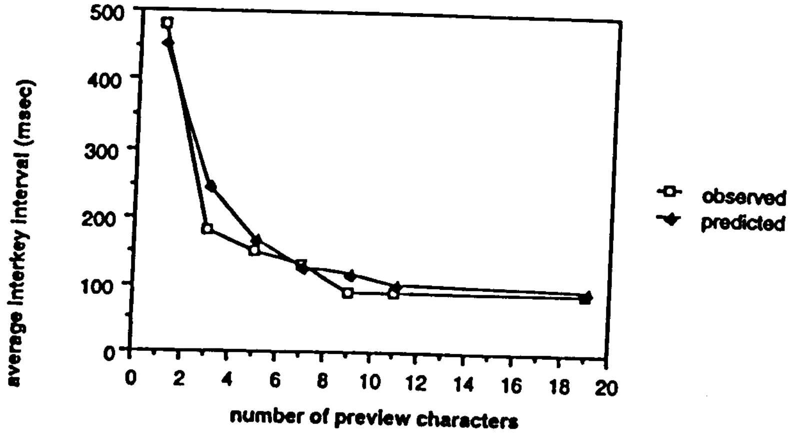


The analysis was done using the example sentence, all of Salthouse's window conditions, and a 120-gwpm typist (Salthouse reports the data for his fastest participant, a 117-gwpm typist). The results of the analysis appear in Figure 7; the average absolute percent error is 15.8%. Thus, this more complex analysis, using the full power of TYPIST, provides quantitative predictions sensitive to the parameter varied in the experiment (viz., the window size).

3.3. Using the Timeline View and Sensitivity to Task Differences

Human performance is often quite sensitive to small changes in the task situation. Using a stopping span task, I demonstrate that TYPIST is also

Figure 7. Restricted preview predicted and observed results.



sensitive to such changes. This task also provides an opportunity to introduce another view of a schedule chart, the *timeline* view.

The stopping span is the amount of material to which the typist is irrevocably committed to typing at any time in the task; it averages one or two keystrokes. Logan (1982) measured this span directly by asking typists to stop typing as soon as they heard an auditory stop signal. He did this in three slightly different experiments. The first experimental task–a time-contingent, discontinuous typing task–was as follows:

> Three-, five-, and seven-letter words were centered on the screen. ... The words were exposed for 1,000 msec, preceded by a fixation point that was exposed for 500 msec and was extinguished immediately before the word appeared. The intertrial interval was 2,000 msec and began as soon as the word was extinguished. ... The stop signal was a 500-msec, 900 Hz tone. ... It was presented at one of four delays (500, 650, 800, and 950 msec) following the onset of the word ... subjects had no visual record of what they typed ... The words within each length condition were balanced for hand repetition and alternation in the keystrokes they required ... (Logan, 1982, p. 780)

To analyze this task, schedule charts were constructed for the perception and typing out of 3-, 5-, and 7-letter words with all possible same-hand, alternate-hand sequences. Imposed on top of these schedules was a perception of the tone (assumed to be 100 msec, estimated from a click-counting experiment; Card et al. 1983, p. 33) and a cognitive operator that recognized the tone to be the stop signal. The start of the perception operator was positioned at the start of the stop signal (500, 650, 800, or 950 msec). The cognitive operator followed immediately and prevented any more characters from being initiated after that point in time. The number of characters typed after the stop signal started was then the number of characters that had been initiated by the cognitive processor before the cognitive operator recognizing the stop signal had begun. The motor operators associated with the initiation cognitive operators then completed the act by typing out the characters. A timeline[9] form of the diagram shows the sequence of events most clearly (See Figure 8). The average stopping span predicted by this analysis is 1.76 characters, 12.1% above the 1.57 characters observed by Logan (1982).

Logan's second experimental task made the stop signal contingent on an event, the typing of a specific character, and was as follows:

> ... the same as in Experiment 1 except that the routine that accepted responses from the keyboard was rewritten to present a stop signal when a prescribed number of keystrokes had been registered (i.e., immediately after the *n*th keystroke). The copy to be typed was the five- and seven-letter words from the first experiment ... The stop signal occurred on 20% of the trails ... at one of four delays (after 1, 2, 3, or 4 keystrokes had been registered). (Logan, 1982, p. 782)

This event-contingent stopping task was analyzed in the same way as the time-contingent task, looking at all possible same- and alternate-hand sequences of five- and seven-letter words and all possible stop signal onsets (after the first, second, third and fourth letters typed). The average predicted stopping span was 1.55 characters, 9.9% above the observed 1.41 characters.

Finally, Logan's (1982) third experiment, also event-contingent, examined stopping behavior within a sentence rather than a single word. The same type of analysis was used for this task and the result was an average predicted stopping span of 2.08 characters, 3.7% below the observed average of 2.16 characters.

Thus, small changes in the typing tasks produce different estimates of the stopping span, just as they produce differences in human behavior (see Figure 9). TYPIST predicts the pattern of results found by manipulating the tasks with an average absolute percent error of 8.6%.

[9] A task timeline is an alternative representation of a schedule chart. Time is represented along the horizontal. The width of the task boxes are proportional to their duration. Tasks that overlap in time appear stacked above each other. The critical path is indicated in the task timeline by boxes that have no striped area at their right side. The striped area is the *slack time* for each subtask, the time that the subtask could be delayed without affecting the duration of the total task. Subtasks on the critical path have no slack time, and thus no striped area.

gun. The motor
ators then com-
n of the diagram
8). The average
12.1% above the

al contingent on
lows:

that accepted
signal when a
mediately after
en-letter words
of the trails ...
en registered).

he same way as
l alternate-hand
op signal onsets
he average pre-
observed 1.41

ntingent, exam-
ngle word. The
was an average
the observed

nt estimates of
n behavior (see
y manipulating

le chart. Time is
proportional to
each other. The
o striped area at
the time that the
al task. Subtasks

Figure 8. **Timeline of a 60-gwpm typist in Logan's (1982) Experiment 1. This instance has a stop signal at 500 msec and a word whose first two letters are one opposite hands and whose third letter is on the same hand as the second.**

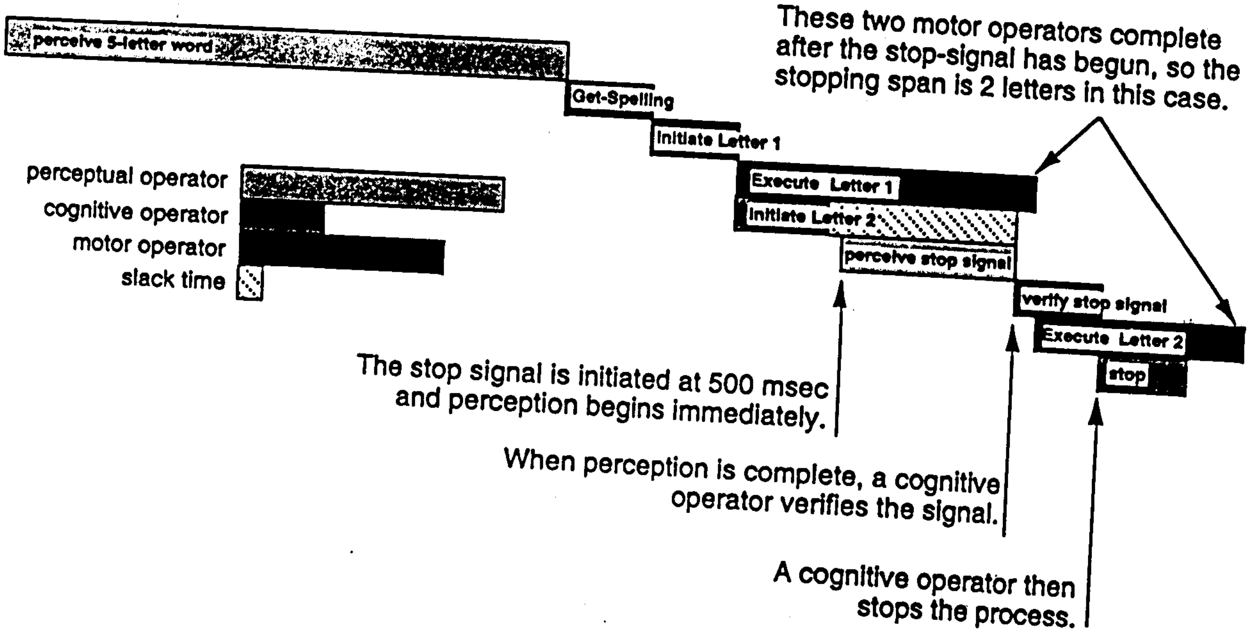

perceive 5-letter word
Get-Spelling
Initiate Letter 1
Execute Letter 1
Initiate Letter 2
perceive stop signal
verify stop signal
Execute Letter 2
stop
perceptual operator
cognitive operator
motor operator
slack time
These two motor operators complete after the stop-signal has begun, so the stopping span is 2 letters in this case.
The stop signal is initiated at 500 msec and perception begins immediately.
When perception is complete, a cognitive operator verifies the signal.
A cognitive operator then stops the process.

Figure 9. Stopping span predictions for three different tasks.

Experiment number	2	1	3
Signal contingency	Event	Time	Event
Context	Single word	Single word	Sentence
Observed stopping span	1.41	1.57	2.16
Predicted stopping span	1.55	1.76	2.08
Percentage error	-9.9	-12.1	3.7

3.4. Integrating TYPIST With Other MHP Models

One of the advantages of building TYPIST within the MHP framework is that it is possible to integrate a model of typing with MHP models of other tasks. Indeed, I argued earlier that it is essential to be able to do so with engineering models because users do not type for typing's own sake, but in service of larger, more complex tasks. Typing often occurs in situations where users are doing other things concurrently, like listening to other people to get the information they are typing (e.g., a credit card number) or visually scanning the screen for information. Other aspects of the task often will interrupt the typing. Although detailed validation of TYPIST's integration into a complex real-world task is beyond the scope of this article, I demonstrate the approach by combining TYPIST with an MHP model of simple RT to account for behavior in a classic dual-task situation.

Salthouse and Saults (1987) gave typists a simple RT task, a simple typing task, and a task where they had to type and perform the RT task concurrently. The results of these tasks are presented in Figure 10. The typing task was not slowed by the concurrent RT task, but the RT task slowed down considerably, indicating that the RT task had not yet become automatic and that the devices used to emphasize the typing task were successful.

The first step in an analysis of this concurrent task situation is to produce an MHP model of the RT task. To do this task, the perceptual processor would perceive the tone with a minimal perceptual operator (100 msec) and place a representation of that tone in WM. The cognitive processor would judge whether this is the right tone or not (this 50-msec verification prevents the model from pressing the foot pedal when somebody sneezes), then initiates the foot press (50 msec).[10] In the absence of

[10] Card et al. (1983) present a similar analysis for a simple RT task without this verification cognitive operator. However, subsequent discussions with Allen Newell led us to include this verification operator in many similar simple tasks, always remembering, however, that sometimes football players *are* pulled offsides (an indication that this verification can sometimes drop out). See Newell (1990) for MHP-like models of several RT tasks that include this verification operator.

3
Event Sentence
2.16
2.08
3.7

els

1HP framework
MHP models of
e able to do so
ing's own sake,
often occurs in
like listening to
., a credit card
Other aspects of
d validation of
yond the scope
YPIST with an
lassic dual-task

task, a simple
rm the RT task
Figure 10. The
ut the RT task
not yet become
ping task were

situation is to
the perceptual
ptual operator
The cognitive
t (this 50-msec
al when some-
the absence of

task without this
with Allen New-
ple tasks, always
led offsides (and
ewell (1990) for
n operator.

Figure 10. **Results of typing and RT tasks (in milliseconds).**

Task	Alone	Concurrent
Typing interkey interval	181	185
SD	64	62
Reaction time	269	431
SD	49	85

data about foot presses, we can estimate that the motor processor executes the foot press with a typical motor operator (70 msec). Figure 11 shows the schedule chart of this task. Because this task is a set of sequential operations, everything is on the critical path and total time predicted for this task is 270 msec, whereas its measured duration was 269 msec.

To model the concurrent task, the model for the simple RT task is superimposed on top of the TYPIST schedule chart for a 60-gwpm typist typing the example sentence (Salthouse and Saults's typists averaged 63.1 net words per minute, nwpm, a measure of speed that corrects for typing errors). The tone was assumed to start at 25 random locations within the example sentence and each location was analyzed in isolation, to simulate the random positioning of the tone within a longer portion of text (Salthouse and Saults used up to 250 words). Because the RT task was not automatic and the typing task was emphasized, the operators that perform the RT task were woven in between those of the typing task, with the typing operators taking precedence. If the perceptual processor was not busy perceiving a word when the tone started, then the perception of the tone began at the onset of the tone; otherwise the perception of the tone began as soon as the perceptual process completed the perception of the word. When the perception of the tone was complete–if the cognitive operator was not busy doing something for the typing task–the verification of the tone began; otherwise, the verification waited until the typing cognitive operation was complete. Then the cognitive operators for the typing task and the RT task were woven together, alternating between tasks if they were competing for cognitive processing time.[11] The motor operator to press the foot pedal began after the foot press was initiated by the cognitive processor and the motor processor was not busy typing a character (Figure 12).

The concurrent RT was predicted from these schedule charts by measuring the time between the randomly assigned onset of the tone and the completion of the foot-press motor operator. The average concurrent RT

[11] Sometimes the typing task involved a same-hand key sequence and the cognitive processor was waiting for the completion of the motor processor. This left enough time for the RT cognitive operators to execute without alternating between tasks.

Figure 11. Schedule chart for the RT task.

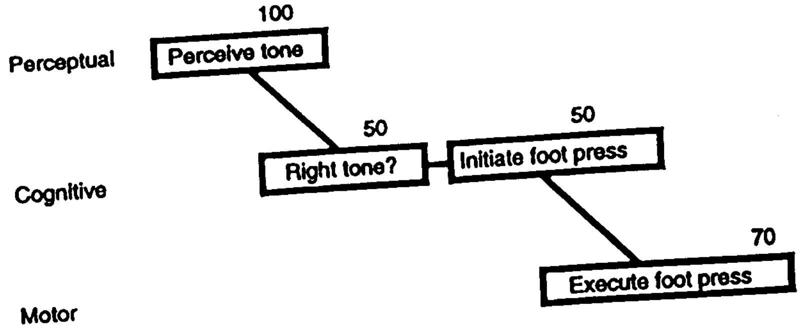
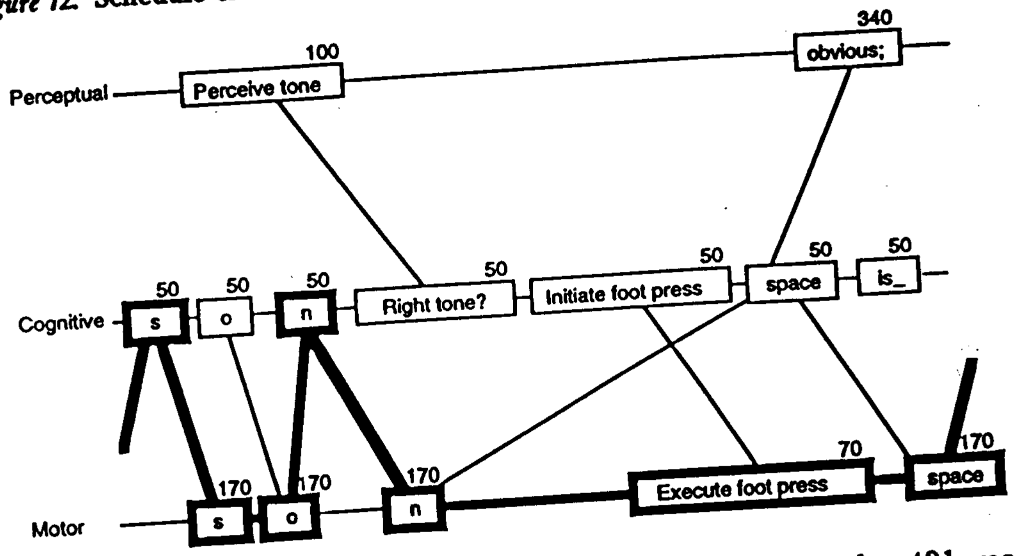


Figure 12. Schedule chart for the concurrent typing and RT tasks.



predicted by this analysis is 435 msec, 0.9% away from the 431 msec observed by Salthouse and Saults (1987).

The effect on the average interkey interval for Salthouse and Saults's (1987) task is reported to be small; the observed mean of the interkey interval was 181 msec for the normal typing task and 185 msec when the concurrent RT task was added. TYPIST predicted an average interkey interval of 195 msec for the normal typing task, based on the typing speed of the participants. The interkeystroke interval for those keystrokes that were interrupted by the RT task was predicted to be 240 msec. However, this was only for those keystrokes that were directly involved in the concurrent task, not the average for the entire concurrent typing task. In the entire concurrent task, there were 30 tones presented within a 1,200-character passage (T. A. Salthouse, personal communication, 1988). Thus, 30 interkeystroke intervals increased to 240 msec and 1,170 keystrokes remained at an average of 195 msec. With this ratio, the overall average interkey interval for typing with the concurrent task was predicted to be

196 msec, almost the same as the typing-alone interkeystroke time. This analysis supports the claim that a concurrent task has little or no effect on the typing speed of an expert typist. An interesting prediction of TYPIST is that the interkeystroke intervals occurring as the foot press is occurring will increase; this prediction is left for future empirical verification.

3.5. Modeling Individual Differences in Strategy With TYPIST

The preceding tasks were extremely simple, with unambiguous instructions to the user. A single reasonable procedure for accomplishing the task falls out of the instructions and the assumptions of TYPIST. The next task is slightly more complex, with more ambiguous instructions. This leaves room for different users to interpret the instructions differently, adopt different strategies to accomplish the task, and display more variable performance. These individual differences can be emulated in TYPIST by creating a different version of the task model for each possible strategy and examining the different performance predictions made.

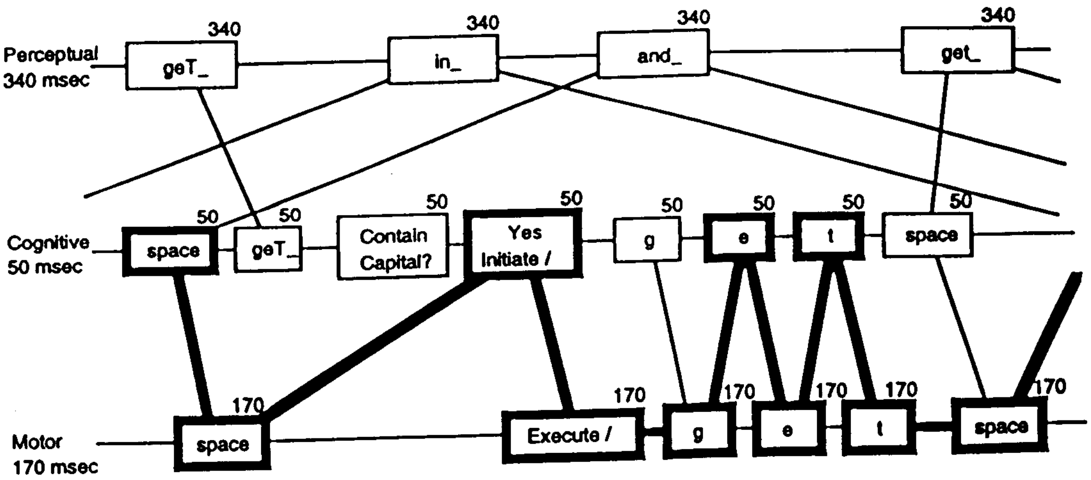
The instructions for this task asked typists to type a slash (/) as soon as they see a capital letter anywhere in the midst of the material they were transcribing from a monitor. This task measures the detection span.

> In this task a large number of characters will always be visible on the display, but occasionally a capital letter will appear. Whenever you notice a capital letter anywhere on the line you should press the '/' key as soon as you can and then resume typing. The capital letters should not be typed as capitals, but whenever you detect an upper-case letter you should press the '/' key. Always try to type as normally as possible. (Salthouse & Saults, 1987, p. 189)

TYPIST can produce three strategies that conform both to the instruction to "try to type as normally as possible" and the "press the '/' key as soon as you can". The first generates the '/' keypress when the word containing it is being typed out and the last two generate the '/' keypress close to when the capital is perceptually encoded.

The first strategy, the *spelling algorithm,* assumes that the perception operator includes an encoding of whether a capital exists in the spelling of the word. When the spelling of the word is brought into WM in preparation for that word being typed, a test is performed on the spelling to see if a capital exists. If a capital does exist, then the result of the test is to type the '/'. Figure 13 reveals that, because the initiation of the preceding space triggers the retrieval of the spelling of the next word, the space is always typed before the '/' is hit. Since the letters of the word containing a capital cannot be typed until the '/' is hit, then the position of the capital letter in the word is the detection span (i.e., if the capital is the first letter, the

Figure 13. **Schedule chart of the spelling algorithm for the detection span task.**



detection span is one; if it is the second, the detection span is two, etc.). With a stimulus material of only 4-letter words, this gives detection spans of 1, 2, 3, and 4 characters.

The *perception algorithm* has two versions: *wait* and *parallel.* The assumption is made that the participants looked ahead as they would in the course of normal typing, but made the judgment about whether a capital existed after the word was perceived normally. The perceptual operator then encoded whether a word included a capital independent of its spelling. The test for a capital was performed as soon as the perception was complete, and the '/' hit if a capital was detected. The difference between the two versions of the algorithm is that in the wait version (Figure 14), the perception of a word is initiated in the course of normal typing, but the cognitive processor waits for the perception to be complete and makes the test before continuing with typing. In the parallel version (Figure 15), the perception proceeds as it does in normal typing but the cognitive processor continues with the typing task until the perception is complete, and then the test is made.

In the wait case (Figure 14), the initiation of a space at the end of a word is the cognitive activity that triggers the perception of the next word, and the decision about the capital waits for the perception to be completed. Because of this, the first character of the word that is two words before the word with the capital will always be the character at the leftmost side of the screen and the detection span will be 10 plus the position of the capital within the word (i.e., 11, 12, 13, or 14). In the parallel case, the critical path analysis (Figure 15) is more complex because the same-hand/alternate-hand pattern of letters influences how many characters are typed during the perceptual process. The detection spans predicted using this algorithm range from 7 to 13 characters.

Figure 14. Schedule chart of the perception-wait algorithm for the detection span task.

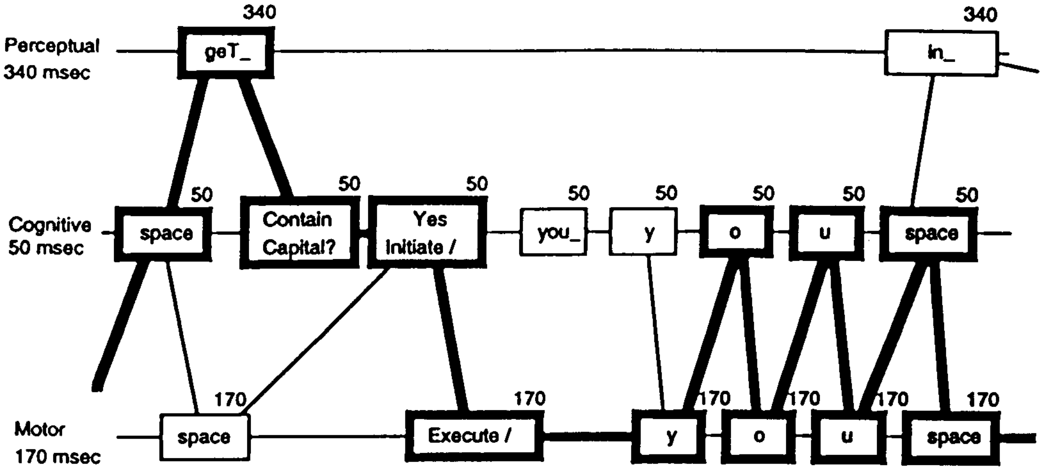
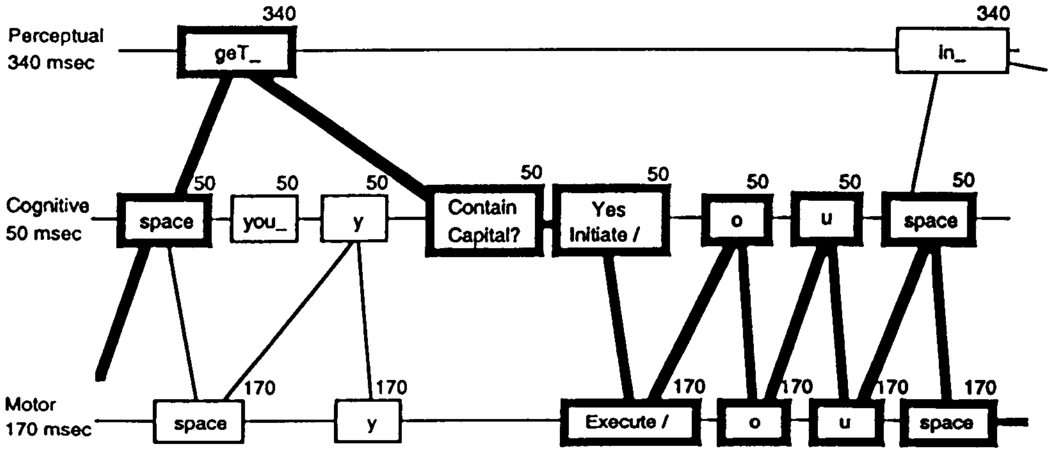


Figure 15. Schedule chart of the perception-parallel algorithm for the detection span task.



An estimated typing speed of 60 wpm (approximately the average of experimental participants), averaged over these three possible algorithms, yielded a predicted detection span of 8.56. Salthouse and Saults (1987) observed a mean detection span of 8.1 characters (*SD* = 4.6 characters) for one study and 7.8 characters (*SD* = 5.0 characters) for another–an average of 7.95 characters.

Salthouse and Saults (1987) comment that the distribution of the observed detection spans were quite flat and ranged from 1 to 15 characters. Great variation and range are also predicted by TYPIST: most values from 1 to 14 fall out of the critical path analyses. To explain these results, Salthouse and Saults proposed that the participants violated the instructions to "type normally" and periodically decided to interrupt typing to scan for capitals. The TYPIST analysis suggests that a true stop-and-scan strategy, like the one proposed by Salthouse and Saults, which assumes a violation of the task instructions, is not necessary to explain individual differences in the data.

3.6. Using TYPIST at Different Levels of Approximation

The assumptions of TYPIST define one level of engineering model of typing performance that differentiates between hands, but not between fingers on a hand. However, this is only one level of approximation and other models have been shown to work well at other levels of approximation. Specifically, the KLM (Card et. al., 1980, 1983) assigns a single interkeystroke estimate to each keystroke, irrespective of which hand types it. At the other end of the spectrum, Rumelhart and Norman (1982) used a connectionist model to represent the detailed overlapping of fingers both between- and within-hand. In this section, I show how TYPIST relates to each of these other levels.

In normal transcription typing, perception stays far ahead of the finger movements; thus the critical path goes through the motor operators alone for alternating-hand sequences and through both the motor operator and the intervening cognitive operator for same-hand sequences (e.g., typing the example sentence in Figure 2). If an analysis situation did not require understanding the perceptual aspect of the task or the interaction of perception, cognition, and motor processes (as all of the typing tasks discussed thus far have required), then the perceptual, cognitive, and motor operators could be aggregated into a single estimate in the analysis, specifically an average time per keystroke could be used. Given TYPIST's estimate of the motor operator (Figure 4) and an assumption that half of the keystrokes will be same-hand and half will be alternate-hand, TYPIST reduces to an overall estimate of 0.20 sec/keystroke for a 60-gwpm typist and 0.13 sec/keystroke for a 90-gwpm typist. Compare this to the estimates used for the KLM: 0.20 sec/keystroke for a 55-wpm typist and 0.12 sec/keystroke for a 90-wpm typist (Card et al., 1983, p. 264). At this level of approximation, these differences are negligible and TYPIST essentially reduces to the KLM.

To examine the other extreme, consider the robust typing phenomenon that the time to hit a key depends on the keys surrounding it.[12] For example, Rumelhart and Norman (1982) found that the time to hit "e" ranges from 159 msec when the preceding character is "r" to 215 msec when the preceding character is "c". TYPIST's Assumption 2 forces each

[12] This phenomenon is clearly true for the character immediately preceding the key, and seems to extend more weakly up to three characters prior to the key (Gentner, 1982, 1983; Shaffer, 1978).

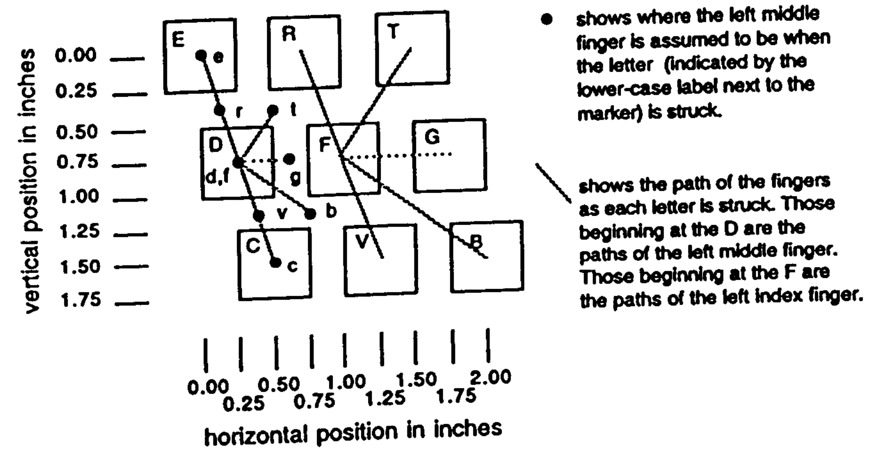
finger to wait until the preceding character is finished to start moving. As discussed when that assumption was presented, this is a simplifying assumption that does not capture the finger-movement overlap that expert typists actually display. It also prevents TYPIST, as presented in Section 2, from making different interkeystroke time predictions depending on the context. However, Assumption 2 can be relaxed, if the analysis situation requires it, in the following way.

Gentner (1982, 1983), Rumelhart and Norman (1982), and Shaffer (1973, 1978) have argued that these time differences can be attributed primarily to the structure of the hand and the topology of the keyboard. To take these aspects of the typist and the task into account in TYPIST, I assume a keyboard with a 0.5-in.-square key, and a 0.75-in. key separation, center-to-center (Figure 16). For same-finger digrams, I assume that the finger has to move the entire distance between keys, center-to-center. For different-finger digrams, I assume that the middle and index fingers stay in the same relative horizontal position as the index finger begins to move to strike a key, but that the middle finger moves only halfway along the trajectory it would take if that relation were strictly maintained. Following this assumption, the positions of the middle finger when the index finger hits another key are marked on Figure 16 with • and the lower-case letter of the key being hit. To hit "e", the middle finger would then have to move from where it was when the index finger hit its key to the center of the *e* key. The straight-line distance that the middle finger has to travel to hit "e" is called the "Distance moved" (Figure 17).

However, the horizontal distanced moved is not sufficient to explain interkeystroke time. After all, hitting the same key twice requires no horizontal movement, but does take some time (165 msec in the case of the *e–e* digram). Logically, the finger must move horizontally to the key, down to hit the key, and up again to clear the other keys in preparation for moving horizontally to the next key. Therefore, at this level of detail, I assume three components to the motor movement after it has been cognitively initiated: horizontal movement to get over the key, down, and up. Our own observations of videotapes of data entry reveal that down and up times are approximately symmetric and of the order of 60 to 100 msec (Gray et al., 1993). Therefore, a simple assumption is that the 165 msec for the *e–e* transition is split equally between the down time and the up time–83 msec each.

At the level of analysis defined by the original TYPIST assumptions some actions (same-hand keystrokes) could not overlap and some (alternate-hand keystrokes) could. At this level of analysis same-finger digrams may not allow overlap but different-finger digrams may. Thus, for same-finger digrams, the total observed time is the sum of the down time (estimated to be 83 msec), the up time (estimated to be 83 msec), and the

Figure 16. **Assumed movement of the left, middle, and index fingers when striking keys on a QWERTY keyboard.**



● shows where the left middle finger is assumed to be when the letter (indicated by the lower-case label next to the marker) is struck.

\ shows the path of the fingers as each letter is struck. Those beginning at the D are the paths of the left middle finger. Those beginning at the F are the paths of the left index finger.

Figure 17. **Predictions of movement time by Rumelhart and Norman's (1982) model and by TYPIST.**

			TYPIST with Fitts's Law			Rumelhart and Norman		
Keys	Total time observed[a]	Distance moved[b]	Estimated move time[a]	Total time predicted[a]	% Error	Simulation units	Total time predicted[a]	% Error
e–e	165	0.00	0	166	0.0	5.9	159	3.6
d–e	201	0.79	34	200	0.5	9.5	192	4.5
c–e	215	1.58	60	226	–1.8	12.7	221	–2.8
r–e	145	0.40	12	145	1.4	7.1	170	–17.2
t–e	159	0.63	26	159	1.2	7.0	169	–6.3
f–e	168	0.79	34	167	0.6	7.3	171	–1.8
g–e	178	0.98	42	175	–0.1	7.2	171	3.9
v–e	178	1.19	49	182	–0.9	7.5	173	2.8
b–e	195	1.35	54	187	–1.4	8.1	179	8.2
Average absolute % error					0.9			5.7

[a] In milliseconds. [b] In inches.

horizontal movement time (the remaining observed time). This horizontal movement time is estimated to be 0 msec for the *e–e* transition, 35 msec for the *d–e* transition, and 60 msec for the *c–e* transition.

Using the movement constraints detailed above, pressing the *f* key leaves the middle finger on the *d* key so the *f–e* transition involves the same horizontal movement time as the *d–e* transition. The down time (83 msec) and the horizontal movement time for the *d–e* transition (35 msec) can be subtracted from the 168 msec total time observed for the *f–e* transition to get an estimate of how much time is spent picking the left index finger up before starting to move the middle finger: 50 msec. This value is less than a pure up time (83 msec), indicating, as Rumelhart and Norman (1982) observe, movements of different fingers overlap. This estimate plus the down time (50 msec + 83 msec = 133 msec) can be subtracted from all the observed times for different finger transitions to get estimates of their horizontal movement times.

A variant of Fitts's Law (Welford, 1968)[13] can be used to calculate the horizontal positioning time from the size of the key (S) and the distance from where the middle finger is assumed to be when the preceding key is hit to the center of that key (D):

$$T_{pos} = I_M \log_2(D/S + 0.5) \tag{1}$$

where I_M is a free parameter. To establish I_M, I used regression between the estimated horizontal movement distance and the calculated horizontal movement times, forcing the regression through zero; the result was 32 msec.

The total time for each interkeystroke time can then be calculated by

$$T_{pred} = 32 \times \log_2(D/S + 0.5) + \text{up time} + \text{down time} \tag{2}$$

where D is the distance moved, S is the size of the key (0.5 in.), up time is 83 msec for same-finger movements and 50 msec for different-finger movements, and down time is 83 msec. The results of this calculation are in Figure 17.

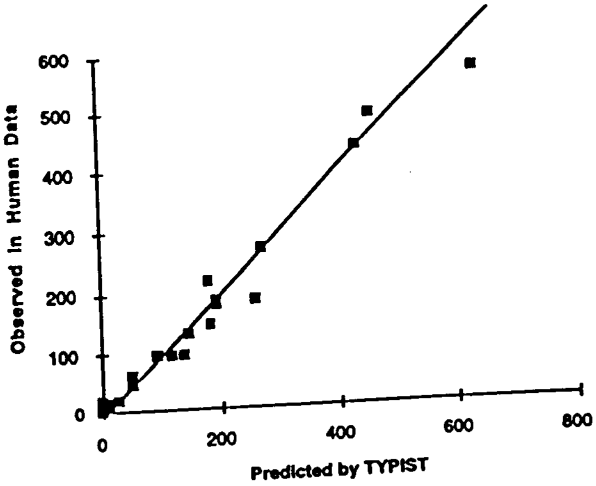
These calculations can be compared to those that can be derived from a connectionist simulation model presented by Rumelhart and Norman (1982). I used regression between their simulation's arbitrary model units and the observed times to establish a slope and intercept for their model. The resulting equation for finding an absolute predicted time from their model is:

$$T_{pred} = 104.5 + 9.2 \times \text{simulation-result} \tag{3}$$

The observed times and the two different predicted times can be found in Figure 17. The percent error for each key transition was calculated for both the predictions; the average absolute percent errors for each prediction technique also appear. The TYPIST-based estimates account for 95% of the variance ($p < .01$), whereas Rumelhart and Norman's (1982) simulation accounts for 69% of the variance ($p < .01$) for these nine digraphs. Rumelhart and Norman report accounting for 74% of the variance on a larger set of 66 most common digraphs in English, but only present the observed times for these nine digraphs.

[13] There is some controversy as to whether Fitts's Law can account for behavior that is not controlled visually and/or has a small Index of Difficulty ($2D/S < 3$). Welford's variant fits the data well, but Gan and Hoffman (1988) believe this may be "fortuitous ... due to the close similarity of the square root and logarithmic relationships." (p. 834). However, for engineering purposes in typing tasks, these fine distinctions may not be important.

Figure 18. **Plot of quantitative predictions made by TYPIST against the observed human performance data for the basic phenomena and units of typing described by Salthouse (1986), plus the Detection Span task (Salthouse & Saults, 1987).**



This simple relaxing of TYPIST's original assumptions about overlapping activities explains the single-character context effect quite well. It does not make true a priori predictions, as we would prefer engineering models to do, because the up, down, and overlap times–as well as the slope of the Fitts's Law curve–had to be determined with reference to the data. If TYPIST were to be extended to this level of detail for routine analyses, the next step would be to obtain independent performance data and stabilize these estimates. However, most typing tasks found in the HCI domain will not require such detail, so this work remains for the future if the need arises.

3.7. Summary

This article has shown that, starting with the assumptions of MHP and a general task analysis of transcription typing, TYPIST can be used to model many different typing tasks. It can be used at different levels of detail from back-of-the-envelope calculations to a version using Fitts's Law to predict interkeypress times for different fingers. It is sensitive to small changes in the task conditions, as is human performance, and can reflect differences in strategy to accomplish the same task. Finally, TYPIST can be combined with other MHP models to account for performance in a dual-task situation, pointing the way to integration into larger, real-world tasks.

In previous works (John, 1988, 1993), TYPIST has been applied exhaustively to tasks that display all the behavioral phenomena identified as robust by Salthouse (1986). It covers most of the phenomena Salthouse called "basic" and "units of typing" (Figures 18 and 19). However, TYPIST

Figure 19. **Salthouse's (1986) list of robust phenomena in transcription typing and qualitative account of TYPIST's predictions.**

	Phenomenon	Predicted by TYPIST?
Basic phenomena		
1.	Typing is faster than choice reaction time	Yes
2.	Typing is slower than reading	Beyond scope
3.	Typing skill and comprehension are independent	Beyond scope
4.	Typing rate is independent of word order	Yes
5.	Typing rate is slower with random letter order	Yes
6.	Typing rate is slower with restricted preview	Yes*
7.	Alternate-hand keystrokes are faster than same-hand keystrokes	Yes
8.	More frequent letter pairs are typed more quickly	Beyond scope
9.	Interkey intervals are independent of word length	Yes
10.	The first keystroke in a word is slower than subsequent keystrokes	Yes[a]
11.	The time for a keystroke is dependent on the specific context	Yes[b]*
12.	A concurrent task does not affect typing	Yes*
Units of typing		
13.	Copying span is 7–40 characters	Yes*
14.	Stopping span is between one and two characters	Yes*
15.	Eye–hand span is between three and eight characters	Yes
16.	Eye–hand span decreases with decreasing meaning	Yes
17.	Replacement span is about three characters	Yes
Post-29	Detection span is about eight characters	Yes*
Errors		
18.	Only a fraction of errors are detectable without reference to the typed copy	Beyond scope
19.	Substitution errors are mostly adjacent keys	Beyond scope
20.	Intrusion errors are mostly short interkey interval	Beyond scope
21.	Omission errors are mostly long interkey interval	Beyond scope
22.	Transposition errors are mostly cross-hand	Beyond scope
Skill effects		
23.	Two-finger digrams improve faster than one-finger digrams	Beyond scope
24.	Tapping rate increases with skill	Yes
25.	Variability decreases with skill	Beyond scope
26.	Eye–hand span increases with skill	Yes[c]
27.	Replacement span increases with skill	Yes[d]
28.	Copy span is dependent on skill	Yes[e]
29.	Stopping span increases with skill	Yes

Note. An asterisk in the far right column indicates that the analysis making this prediction is included in this article. All predictions can be found in John (1988, 1993).
[a] For very fast typists. [b] With Fitts's Law extension. [c] But about 40% too high. [d] But about 50% too low. [e] But in opposite direction.

makes no predictions about the error commission and detection phenomena identified by Salthouse. An examination of those phenomena with relation to TYPIST, indicates that predicting error-commission phenomena would require a more detailed model of the motor processor than the MHP currently uses. Likewise, predicting the error-detection phenomena would require a more detailed model of the perceptual processor than the MHP currently includes. Both of these directions seem fruitful for future research, and indeed, are currently being investigated in the Executive Process-Interactive Control (EPIC) architecture (Kieras & Meyer, 1994). TYPIST makes rather inaccurate predictions about skill effects. This is probably because the measurement of skill effects is drawn from a broader range of skill (e.g., 20-120-wpm typists) than the range that TYPIST models (60-120 wpm). It is unclear from the published data whether the reported skill effects hold up in this more expert range.

4. CONCLUSION

TYPIST is a member of the class of models that can make a priori quantitative predictions of expert, error-free behavior. Within this scope–given the average typing speed of users–it predicts all of the robust spans of typing (copying span, stopping span, eye–hand span, replacement span, and detection span) to within 20% of their empirically measured values. TYPIST also predicts 9 of the 12 "basic phenomena" identified by Salthouse (1986), failing only on those phenomena that fall outside the mechanisms it posits (e.g., the relationship of typing to reading, comprehension, and the frequency of words in the language). Because TYPIST is built within the MHP, it has also been used as the basis for modeling a dual task where typing is interrupted by a RT task.

The potential for integration with other activities is probably the most important aspect of TYPIST. Pure typing speed is the simplest and perhaps most useful model of typing–when typing is the only thing the person is doing (as might have been the case in "typing pools" of old)–but a more integrated modeling technique will be needed for more complex tasks. An extended version of TYPIST and the MHP, called CPM-GOMS (John, 1990), is applicable to tasks that involve visual search (e.g., Chuah, John, & Pane, 1994), auditory input, simple decision-making, routine conversation, using a mouse, and typing. This has proved to be a useful tool for analyzing expert performance on conventional workstations where parallel activities are necessary (Gray et al., 1993). We expect that emerging multimodal input and output systems will benefit even more from models such as these.

NOTES

Acknowledgments. I thank Allen Newell, John R. Anderson, and Lynne M. Reder for their guidance and contributions to this work. Also, the detailed comments of David Rosenbaum, Timothy Salthouse, Richard Young, and several anonymous reviewers contributed substantially to the readability of this article.

Support. This research was supported in part by the Office of Naval Research, Perceptual Science Program (Contract No. N00014-87-K-0432) and the Cognitive Science Program (Contract No. N00014-89-J-1975N158). The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research or the United States Government.

Author's Present Address. Bonnie E. John, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15213. E-mail: bej@cs.cmu.edu.

HCI Editorial Record. First manuscript received January 8, 1996. Revision received May 5, 1996. Accepted by Richard Pew. Final manuscript received July 1, 1996. – *Editor*

REFERENCES

Card, S. K., Moran, T. P., & Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM, 23,* 396–410.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human–computer interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Chuah, M. C., John, B. E., & Pane, J. (1994). Analyzing graphic and textual layouts with GOMS: Results of a preliminary analysis. *Proceedings Companion of the CHI'94 Conference on Human Factors in Computing Systems,* 323–324. New York: ACM.

Cooper, W. E. (1983). *Cognitive aspects of skilled typewriting.* New York: Springer-Verlag.

Gan, K., & Hoffman, E. R. (1988). Geometrical conditions for ballistic and visually controlled movements. *Ergonomics, 31,* 829–839.

Gentner, D. R. (1982). Evidence against a centrol control model of timing in typing. *Journal of Experimental Psychology: Human Perception and Performance, 8,* 793–810.

Gentner, D. R. (1983). The acquisition of typewriting skill. *Acta Psychologica, 54,* 233–248.

Gentner, D. R. (1987). Timing of skilled motor performance: Tests of the proportional duration model. *Psychological Review, 94,* 255–276.

Gray, W. D., John, B. E., & Atwood, M. E. (1993). Project Ernestine: Validating a GOMS analysis for predicting and explaining real-world task performance. *Human–Computer Interaction, 8,* 237–309.

Guinness book of records. (1995). New York: Facts on File.

Heath, R. A., & Willcox, C. H. (1990). A stochastic model for inter-keypress times in a typing task. *Acta Psychologica, 75,* 13–39.

Hershman, R. L., & Hillix, W. A. (1965). Data processing in typing: Typing as a function of kind of material and amount exposed. *Human Factors, 7,* 483–292.

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

John, B. E. (1990). Extensions of GOMS analyses to expert performance requiring perception of dynamic visual and auditory information. *Proceedings of the CHI'90 Conference on Human Factors in Computing Systems,* 107–115. New York: ACM.

John, B. E. (1993). *A quantitative model of expert transcription typing* (Tech. Rep. No. CMU-CS-93-120). Pittsburgh, PA: Carnegie Mellon University, Computer Science Department.

John, B. E., & Newell, A. (1990). Stimulus-response compatibility in a unified theory of cognition. In R. W. Proctor & T. G. Reeve (Eds.), *Stimulus-response compatibility: An integrated approach* (pp. 427-479). Amsterdam: Elsevier.

Kieras, D. E., & Meyer, D. E. (1994). *The EPIC architecture for modeling human information-processing: A brief introduction* (EPIC Tech. Rep. No. 1, TR-94/ONR-EPIC-1). Ann Arbor: University of Michigan, Department of Electrical Engineering and Computer Science.

Logan, G. D. (1982). On the ability to inhibit complex movements: A stop-signal study of typewriting. *Journal of Experimental Psychology: Human Perception and Performance, 8,* 778-792.

Logan, G. D. (1983). Time, information, and the various spans in typewriting. In W. E. Cooper (Ed.), *Cognitive aspects of skilled typewriting* (pp. 197-224). New York: Springer-Verlag.

McClelland, J. L. (1979). On the time-relations of mental processes: An examination of systems processes in cascade. *Psychological Review, 86,* 287-330.

McLeod, P., & Hume, M. (1994). Overlapping mental operations in serial performance with preview: Typing. A reply to Pashler. *Quarterly Journal of Experimental Psychology, 47A,* 193-199.

Nelson, G. H., Lehman, J. F., & John, B. E. (1994). Integrating cognitive capabilities in a real-time task. *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society,* 353-358. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

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

Ostry, D. J. (1983). Determinants of interkey times in typing. In W. E. Cooper (Ed.), *Cognitive aspects of skilled typewriting* (pp. 225-246). New York: Springer-Verlag.

Pashler, H. (1994a). Overlapping mental operations in serial performance with preview: Typing (Comment on McLeod and Hume). *Quarterly Journal of Experimental Psychology, 47A,* 201-205.

Pashler, H. (1994b). Overlapping mental operations in serial performance with preview. *Quarterly Journal of Experimental Psychology, 47A,* 161-191.

Rothkopf, E. Z. (1980). Copying span as a measure of the information burden in written language. *Journal of Verbal Learning and Verbal Behavior, 19,* 562-572.

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel ditributed processing: Explorations in the microstructure if cognition* (Vol. 1). Cambridge, MA: MIT Press.

Rumelhart, D. E., & Norman, D. A. (1982). Simulating a skilled typist: A study of skilled cognitive-motor performance. *Cognitive Science, 6,* 1-36.

Salthouse, T. A. (1984a). Effects of age and skill in typing. *Journal of Experimental Psychology: General, 113,* 345-371.

Salthouse, T. A. (1984b). The skill of typing. *Scientific American, 250,* 128-135.

Salthouse, T. A. (1985). Anticipatory processes in transcription typing. *Journal of Applied Psychology, 70,* 264-271.

Salthouse, T. A. (1986). Perceptual, cognitive, and motoric aspects of transcription typing. *Psychological Bulletin, 99,* 303-319.

Salthouse, T. A., & Saults, J. S. (1987). Multiple spans in transcription typing. *Journal of Applied Psychology,* 72, 187-196.

Schweickert, R. (1978). A critical path generalization of the additive factor method: Analysis of a Stroop task. *Journal of Mathematical Psychology, 18,* 105–139.

Schweickert, R. (1980). Critical path scheduling of mental processes in a dual task. *Science, 209,* 704–706.

Shaffer, L. H. (1973). Latency mechanisms in transcription. In S. Kornblum (Ed.), *Attention and performance IV* (pp. 435–446). New York: Academic.

Shaffer, L. H. (1975). Control processes in typing. *Quarterly Journal of Experimental Psychology, 27,* 419–432.

Shaffer, L. H. (1978). Timing in the motor programming of typing. *Quarterly Journal of Experimental Psychology, 30,* 333–345.

Shaffer, L. H., & French, A. (1971). Coding factors in transcription. *Quarterly Journal of Experimental Psychology, 23,* 268–274.

Shaffer, L. H., & Hardwick, J. (1970). The basis of transcription skill. *Journal of Experimental Psychology, 84,* 424–440.

Sternberg, S., Knoll, R. L., & Wright, C. E. (1978). Experiments on temporal aspects of keyboard entry. In J. P. Duncanson (Ed.), *Getting it together: Research and applications in human factors* (pp. 28–50). Santa Monica, CA: Human Factors Society.

Stires, D. M., & Murphy, M. M. (1962). *PERT (Program Evaluation and Review Technique) CPM (Critical Path Method).* Boston: Materials Management Institute.

Thomas, E. A., & Jones, R. G. (1970). A model for subjective grouping in typewriting. *Quarterly Journal of Experimental Psychology, 22,* 353–367.

Webster's new collegiate dictionary. (1977). Springfield, MA: G. & C. Merriam Company.

Welford, A. T. (1968). *Fundamentals of skill.* London: Methuen.

HUMAN–COMPUTER INTERACTION

A Journal of Theoretical, Empirical and Methodological Issues of User Science and of System Design

Volume 11, Number 4
1996

LEA LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS
Mahwah, New Jersey

HUMAN-COMPUTER INTERACTION

EDITOR: Thomas P. Moran, *Xerox Palo Alto Research Center*

ADMINISTRATIVE EDITOR: Patricia Sheehan, *Xerox Palo Alto Research Center*
PRODUCTION EDITOR: Joseph M. Coda, *Lawrence Erlbaum Associates, Inc.*

Human-Computer Interaction is published quarterly by Lawrence Erlbaum Associates, Inc., 10 Industrial Avenue, Mahwah, NJ 07430-2262.

Subscriptions (based on one volume per calendar year): Institutional, $215; Individual, $39. Subscriptions outside the United States and Canada: Institutional, $245; Individual, $69. Claims for missing issues cannot be honored beyond 4 months after mailing date. Duplicate copies cannot be sent to replace issues not delivered due to failure to notify publisher of change of address.

This journal is abstracted or indexed in: *Cambridge Scientific Abstracts: Health and Safety, Science Abstracts*; *Computer Abstracts*; *Educational Technology Abstracts*; *Engineering Information, Inc.*; *Ergonomics Abstracts*; *ERIC Clearinghouse in Education*; *ISI: SciSearch, Research Alert, Current Contents/Engineering, Computers & Technology*; *Knowledge Engineering Review*; *Library and Information Science Abstracts*; and *PsycINFO/Psychological Abstracts.* Microform copies of this journal are available through UMI, Periodical Check-In, North Zeeb Road, P.O. Box 1346, Ann Arbor, MI 48106-1346.

Special requests for permission should be addressed to the Permissions Department, Lawrence Erlbaum Associates, Inc., 10 Industrial Avenue, Mahwah, NJ 07430-2262.

Printed in the United States

ISSN 0737-0024

HUMAN–COMPUTER INTERACTION

EDITOR: Thomas P. Moran, *Xerox Palo Alto Research Center*

ADMINISTRATIVE EDITOR: Patricia Sheehan, *Xerox Palo Alto Research Center*
PRODUCTION EDITOR: Joseph M. Coda, *Lawrence Erlbaum Associates, Inc.*

Human–Computer Interaction is published quarterly by Lawrence Erlbaum Associates, Inc., 10 Industrial Avenue, Mahwah, NJ 07430-2262.

Subscriptions (based on one volume per calendar year): Institutional, $215; Individual, $39. Subscriptions outside the United States and Canada: Institutional, $245; Individual, $69. Claims for missing issues cannot be honored beyond 4 months after mailing date. Duplicate copies cannot be sent to replace issues not delivered due to failure to notify publisher of change of address.

This journal is abstracted or indexed in: *Cambridge Scientific Abstracts: Health and Safety, Science Abstracts*; *Computer Abstracts*; *Educational Technology Abstracts*; *Engineering Information, Inc.*; *Ergonomics Abstracts*; *ERIC Clearinghouse in Education*; *ISI: SciSearch, Research Alert, Current Contents/Engineering, Computers & Technology*; *Knowledge Engineering Review*; *Library and Information Science Abstracts*; and *PsycINFO/Psychological Abstracts.* Microform copies of this journal are available through UMI, Periodical Check-In, North Zeeb Road, P.O. Box 1346, Ann Arbor, MI 48106–1346.

Printed in the United States

ISSN 0737–0024