Toward Expert Typing in ACT-R

Robert St. Amant (stamant@csc.ncsu.edu), Prairie Rose Goodwin (prgoodwi@ncsu.edu), Ignacio X. Domínguez (ignacioxd@ncsu.edu), and David L. Roberts (robertsd@csc.ncsu.edu)
Computer Science Department
North Carolina State University
Raleigh, NC 53706, USA

Abstract

This paper describes an effort to integrate the TYPIST theory of expert transcription typing into the ACT-R cognitive architecture. Our goal is to strike a reasonable balance between a match to the highly accurate predictions of TYPIST and the architectural constraints imposed by ACT-R. The model we have built provides good predictions of human performance on most basic typing phenomena, though less accurately than TYPIST. We present the design of the model, a description of software to support model execution and experimentation, and the results of performance tests comparing the model's predictions with human typing data in the literature.

Keywords: Cognitive modeling; ACT-R; TYPIST; transcription typing

Introduction

TYPIST (John, 1996) is a theory of transcription typing based on the Model Human Processor (MHP) (Card, Moran, & Newell, 1983). In this paper we describe an attempt to incorporate the TYPIST theory into ACT-R.

The ACT-R model of typing we have developed gives predictions qualitatively consistent with TYPIST (though typically with lower accuracy) for fourteen typing phenomena identified by Salthouse (1986). The first contribution of this paper is an ACT-R model that reflects the basic control structure of TYPIST, with some pragmatic modifications to accommodate differences between ACT-R and the MHP. Such a model exists internal to CogTool (John, Prevas, Salvucci, & Koedinger, 2004), and typing models exist in other architectures (e.g., QN-MHP (Wu & Liu, 2004)), but to our knowledge this is the first "standalone" ACT-R model of typing. The second contribution is software to support execution and evaluation of the task of transcription typing, which may be useful in other contexts. The third contribution is a set of basic typing performance results (Salthouse, 1986). Our model does not (yet) offer new insights into typing, but it extends the scope of modeling possible with ACT-R and it has helped us identify new directions for architectural work.

An ACT-R typing model

We begin with an outline of TYPIST. John (1996, p. 326) summarizes the basic method as follows:

> TYPIST perceives a *chunk*... 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... If a letter is perceived alone, then the letter is initiated immediately following the perception and executed.

The perceptual, cognitive, and motor processors of the MHP work in parallel, while operators internal to each processor are executed sequentially; data flow requirements impose constraints on operators across the processors. For example, a cognitive operator can act on a word in working memory (WM) only after it has been made available by a perceptual operator, and motor operators to type the characters in the word can execute only after its spelling has been retrieved by a cognitive operator.

TYPIST's WM has a limited capacity. It can store up to three chunks of text that have not yet been processed by cognitive operators. This capacity constrains the perceptual processor in its ability to look ahead at words to be typed.

At a conceptual level, the ACT-R model works similarly. The biggest difference is that single-character chunks are treated the same as multi-character chunks in the requirement that their spelling be retrieved from memory. This simplification was made to limit the complexity of the model.

We describe the structure of the model mainly in terms of processing words, but the model processes at the level of syllables and characters as well. Our goals in modeling TYPIST included reproducing its structure as well as its performance, while minimizing changes to the ACT-R architecture and the parameter settings used in model execution.

In automatic pre-processing for a typing trial, a sentence is first decomposed into words separated by spaces; punctuation is treated as part of a word. Each word is then decomposed either into syllables or a combination of trigrams and bigrams, using a left-to-right greedy algorithm. Each word or syllable decomposition is stored in a *spelling* chunk in declarative memory, with an ordered set of slots $c_1 \dots c_n$ containing characters. For modeling convenience we assume that characters are not context sensitive, and are processed as individual elements like an array. However, priming studies indicate that skilled typists perceive characters as order-dependent and are thus chained together more like a linked list (Snyder & Logan, 2014).

When the model begins executing, the goal buffer is loaded with a *typing* chunk with slots to maintain the previous visual location, the current state of visual processing, the word to be read, spelled, or typed, along with its visual location, and two variable slots for the current character in the word being typed and a one-character lookahead. In other words, the typing chunk maintains state information for perceptual, cognitive, and motor processing.

A *find-next* and an *attend-next* production perform visual processing following ACT-R modeling conventions. When a word becomes available in the visual buffer, it is recorded by an *add-to-WM* production in "working memory" (discussed below) in two ways: it is stored in the typing chunk and combined with the previously read word in a *previous/next* chunk, to record sequencing. This chunk is added to declarative memory through the imaginal buffer.

The production *initiate-word* retrieves a previous/next chunk from memory, with the word most recently typed being the previous element. (As a special case, *initiate-first-word* fires for the very first word to be typed, which does not have a predecessor.) The *spell* production then retrieves the spelling of this word from declarative memory. The spelling chunk becomes available and is maintained in the retrieval buffer. The *char-slot* in the typing chunk is set to c_1.

Several productions initiate motor actions to type individual characters under different conditions. The basic *initiate-letter* makes a request to the motor module for the current character and advances to the next character (i.e., the typing char-slot is modified from c_i to c_{i+1}). When the current character is a space, *initiate-last-letter-in-word* fires instead, performing the same function and also requesting the retrieval of the next word to be typed. The production *initiate-last-letter-in-syllable* behaves the same way, except that it fires when a special end-of-syllable marker is encountered in the one-character lookahead. Finally, *initiate-single-letter-in-syllable* is used for single-character syllables. A new keystroke can be initiated after the preparation stage of the previous keystroke is complete, following Salthouse (1986): "[I]t is assumed that the typist is executing one keystroke while simultaneously preparing the movement patterns for the next keystroke…" We discuss motor issues in more detail later in this paper.

The ACT-R model inevitably differs from TYPIST, due to the level of modeling detail. Managing data flow dependencies is complex. Some state information is in the form of the status of buffers, but a number of flags are needed in the ACT-R model to ensure proper ordering in production firing. Productions explicitly manage memory: add-to-WM transfers a word from the visual buffer to the typing chunk and creates a new previous/next chunk in memory; initiate-word retrieves the next word to type. Visual processing is also more complex, in particular when a limited preview of text is available. Further, ACT-R visual operations take cognitive processing time, which introduces additional time at word boundaries.

Finally, TYPIST's WM capacity is not simple to reproduce in ACT-R. Without this limit, the ACT-R model looks too far ahead of its keystrokes. An ad hoc solution was implemented in the model: a count is kept such that visual processing is never more than three words ahead of cognitive processing.

Some practical limitations apply to the model. Visual processing in the model assumes that the text of the sentence is on one arbitrarily long line; there is no mechanism to move attention back to the left at the end of a line. Uppercase letters are not handled, automatically being translated to lowercase.

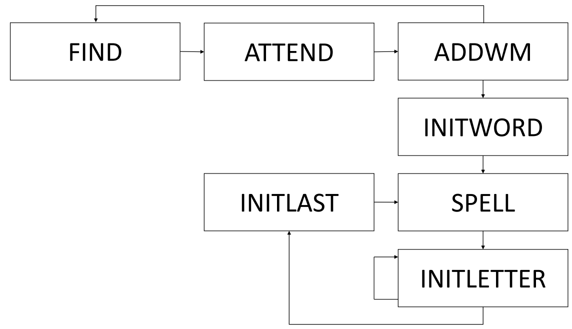

Figure 1: Graphical representation of the typing model.

Modeling support

New software was needed for model execution and experimentation. TYPIST processes text at the level of words, syllables, or letters. To support syllable functionality, we compiled a database containing the syllable decomposition of 166,000 common words. If a word is not in the database, it can be automatically broken down into trigrams and bigrams (those occurring at least 1% as frequently as the most common trigram or bigram in English) and individual letters.

The typing model depends on a small set of motor extensions to ACT-R, including a new movement style, *TYPIST-hit-key*. A finger can move to non-integer $\langle x, y \rangle$ locations, and unlike *press-key*, the finger pressing a key does not return to the home row afterwards. Further, at high typing speeds, the starting point for finger movement in a future scheduled keypress is not the current finger position; the hand/finger representation was modified to handle this possibility. Other modifications are described in the context of typing phenomena they are intended to support.

A virtual typing window displays the text to be typed, in a single line. For ease of experimentation, the window maintains text at both the word and syllable level, providing a model with either as determined by experimental settings. Variations on this window were developed to support different tests as described later in this paper.

A new virtual keyboard for typing was developed, duplicating the layout of the keyboard used for some tests of TYP-

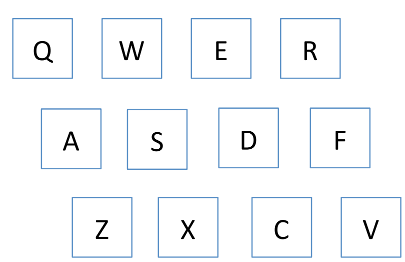

Figure 2: Portion of the typing keyboard layout.

IST (John, 1996, Fig. 16). A portion is shown in Figure 2. The typing keyboard is incomplete, including only letters, a space, a semicolon, a comma, a period, and a slash. In use, timing is very similar to the default keyboard.

Model performance

Some of John's discussion of TYPIST relies on an example sentence: "One reason is quite obvious; you can get in and out without waiting for the elevator." We used this sentence to determine basic timing for the ACT-R model as well as in some testing, as described below. With the main motor timing parameters (preparation, initialization, and burst time) at their default settings, the ACT-R model types the elevator sentence at a rate of 48 words per minute (wpm), with a mean interkey interval of 232 ms. Most of the typing tests we consider are for a 60 wpm typing rate, which can be achieved by changing feature preparation time from 50 ms to 40 ms, producing a mean interval of 207 ms. The model reaches its maximum speed of 108 wpm when these three motor parameters and the Fitts' Law coefficient are set to zero (following John's assumption that variability in typing speed is due to motor speed). TYPIST's maximum speed is 180 wpm.

TYPIST was evaluated with respect to 29 typing phenomena identified by Salthouse (1986). Twelve of these are considered basic phenomena; five concern units of typing, five errors, and the remainder skill effects (John, 1996). For many of these phenomena, TYPIST was evaluated by multiple tests. The evaluation of the ACT-R typing model is much less extensive: the basic phenomena and two units of typing phenomena are examined, and a single test is used in each case. All of the tests used to evaluate the ACT-R model were originally used for TYPIST.

Phenomenon 1. Typing is faster than choice reaction time. Salthouse (1984) describes an experiment in which participants see a stimulus (an L or an R) and type the leftmost or rightmost character key on the bottom row of the keyboard (a Z or a slash). Following the approach outlined for TYPIST, we developed a reduced model that does not store ordering or spelling information, with two new productions to map correctly between the characters. We also modified the typing window such that its contents update to a new character after each keystroke. A comparison between mean interkey interval and reaction time observed in Salthouse's experiment, TYPIST's predictions, and the ACT-R model's predictions are shown in the table below. The ACT-R numbers are from a sample run using a random string of Ls and Rs, with the baseline mean interval produced by the model typing the string as if it were a single word. Absolute errors, as a percentage of observed values, are given in parentheses.

Statistic	Target	TYPIST	ACT-R
Interval (ms)	177	195 (10.2%)	190 (7.6%)
RT (ms)	560	635 (13.4%)	505 (9.9%)

Phenomenon 2. Typing is slower than reading.

Phenomenon 3. Typing skill and comprehension are independent. These are beyond the scope of TYPIST and are not implemented in the ACT-R model.

Phenomenon 4. Typing rate is independent of word order. As with TYPIST, the treatment of words by the ACT-R model does not depend on their order of appearance. Salthouse cites a loss of 2.8% between meaningful sentences and randomly arranged words. In the ACT-R model, no differences in words per minute are seen with re-ordered words in random sentences, generated by sampling from the word database.

Phenomenon 5. Typing rate is slower with random letter order. West and Sabban (1982) describes an experiment in which participants typed easy prose sentences (EP, e.g., "I have your letter in which you ask about the prices"), sentences in which "words" were constructed by rearranging word parts but retaining the ordering of the letters (LC, letter combinations, e.g., "I veha uryo terlet ni chwhi ouy ska outab eth espric"), and sentences in which the ordering of letters in words was arbitrary (LJ, letter jumbles, e.g., "I evah uoyr rtleet ni hcihw oyu ska auobt teh rpcsei"). West and Sabban measure the percent speed increase from LJ to EP, LC to EP, and LJ to LC.

The model applies a single strategy in decomposing words: to syllables (the default behavior) for EP; to common trigrams and bigrams (the default when words are not recognized) for LC; and to individual letters for LJ (explicitly induced). For typists in the range of 55 to 69 wpm, the closest match to the model's 60 wpm, the model performs poorly, though the rank ordering of mean keystroke interval per condition is correctly predicted (EP, 231 ms; LC, 246 ms; LJ, 404 ms). The ratios would be close to observed behavior if the model's LC interval were 40% higher. TYPIST does much better, including different strategies for breaking LC and LJ words down, with an average error of 18% for one plausible combination of strategies across different typing speeds (which means that our results, limited to 60 wpm, are not directly comparable).

Statistic	Target	ACT-R
LJ-EP increase	0.677	0.748 (10.5%)
LC-EP increase	0.416	0.064 (84.6%)
LJ-LC increase	0.187	0.643 (243.7%)

Phenomenon 6. Typing rate is slower with restricted preview. Salthouse (1984) presented typists with a sentence from 60 to 83 characters on a single line. Only a preview of *n* characters for the entire sentence was displayed, with each keystroke causing the preview to advance by sliding the text leftward, removing the first character and adding a new one at the end. Preview sizes were 19, 11, 9, 7, 5, 3, and 1 character. The sentences used are not given by Salthouse; our testing substitutes the elevator sentence. We implemented a preview window that acts approximately the same way, except for the leftward movement of the text. Existing text remains on the screen as well but visual processing is strictly left to right. On

each keystroke, new text may be available to the model; the details of incremental visual processing follow that of TYPIST (John, 1996, p. 335), providing words, syllables, or characters, depending on available space given the size of the preview. TYPIST predicts performance on the elevator sentence (85 characters) at 120 gross wpm, with an error of 15.8%.

Performance of the ACT-R model on this sentence at 60 wpm, in terms of the mean interkeystroke interval in ms, is shown in the table below and in Figure 3, compared with the median interkeystroke interval in the experiment cited above (Salthouse, 1984, Table 2). Excluding the non-preview data, ACT-R model predicts Salthouse's observed data with a mean absolute error of 23.8% and $R^2 = 0.992$.

Preview	Target	ACT-R
None	181	207 (14.2%)
19	179	206 (15.2%)
11	183	214 (17.3%)
9	180	232 (28.0%)
7	185	243 (31.5%)
5	205	288 (40.7%)
3	293	381 (30.2%)
1	645	723 (12.1%)

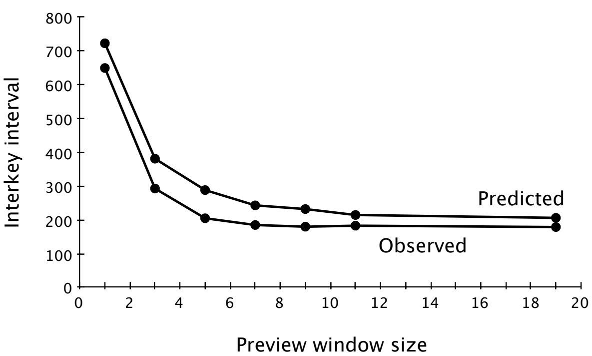

Figure 3: Phenomenon 6: Typing with a preview window

Phenomenon 7. Alternate-hand keystrokes are faster than same-hand keystrokes. Salthouse reports that successive same-hand keystrokes are slower than alternate-hand keystrokes, in the range of 30–60 ms. TYPIST predicts a 50 ms difference, for an error of 11.1% compared with the 45 ms midpoint, by including an additional cognitive operator for same-hand key sequences.

In ACT-R, alternate-hand keystrokes would ordinarily be *slower* than same-hand keystrokes due to feature preparation time. Adopting the solution in TYPIST would require the mapping between keys and hands to be explicit in the model's productions, significantly increasing its complexity. Instead, the function for computing preparation time in the ACT-R motor module was modified to produce an appropriate difference in the opposite direction. On random sentences, the ACT-R model predicts a duration of 47.5 ms, error 5.6%.

Phenomenon 8. More frequent letter pairs are typed more quickly. This is beyond the scope of TYPIST; John observes that it is a small effect, 4% of the variability after same- versus alternate-hand sequences. For the ACT-R model, a test of random sentences shows a near-zero correlation between the relative frequency of a bigram and the model's predicted duration, for the 265 most common English bigrams. These most common bigrams are typed approximately 2.5% faster than the remaining bigrams, because they are more frequently chunked as part of the syllables recognized by the model.

Phenomenon 9. Interkey intervals are independent of word length. Specifically, the duration of the first keystroke in a word is not dependent on its length, and neither are the durations of other keystrokes in the word. Both TYPIST and the ACT-R model show such independence.

Phenomenon 10. The first keystroke in a word is slower than subsequent keystrokes. Salthouse reports a 20% increase for an average typist; TYPIST predicts increases of 0% to 8.4% over typing speeds from 60 wpm to 120 wpm, when the spelling operator is on the critical path. On the elevator sentence, the ACT-R model predicts an increase of 18.5%, for a 7.7% error. On samples of random sentences, the ACT-R model predicts an increase of 18%, for an 11.4% error.

Phenomenon 11. The time for a keystroke is dependent on the specific context. Here, context means that in a sequence of keystrokes, α-β, the duration of β depends on α. TYPIST can predict key context effects with two refinements, based on John's analysis of a set of digraphs e-e, d-e, c-e, r-e, t-e, f-e, g-e, v-e, and b-e (Rumelhart & Norman, 1982). Both refinements decompose a keystroke into horizontal movement of a finger across the keyboard (*move*), key down (β_{down}), then key up (β_{up}). The first is that in the digraph α-β, $\alpha_{up} = 83$ ms for a same-hand, same-finger sequence, 50 ms for a same-hand, different-finger sequence. The second is a relaxation of the MHP assumption of strictly sequential finger movements; John derives a formula for very accurate prediction of the *move* component in these digraphs. Specifically, when the middle finger moves to press a key, the index finger moves part of the distance in the same direction, and vice versa.

TYPIST predicts performance on the digraphs with 0.9% mean absolute error (MAE) and $R^2 = 0.95$. The predictions of the ACT-R model, which does not incorporate John's analysis, have a MAE of 49.8%, $R^2 = 0.211, p = 0.21$.

Digraph	Target	TYPIST	ACT-R
e-e	165	166 (0.0%)	231 (40.0%)
d-e	201	200 (0.5%)	289 (43.8%)
c-e	215	226 (1.8%)	321 (49.3%)
r-e	145	143 (1.4%)	260 (79.3%)
t-e	159	157 (1.2%)	249 (56.6%)
f-e	168	167 (0.6%)	289 (72.0%)
g-e	178	175 (0.1%)	264 (48.3%)
v-e	178	182 (0.9%)	260 (46.1%)
b-e	195	187 (1.4%)	220 (12.8%)

To see why this is a challenge, note the duration predictions for the v-e and b-e key digraphs, as an example: the predicted order is the reverse of the observed. The partial traces below show why. The production initiate-letter fires at the same times for both digraphs (as marked by +), once preparation for the previous keystroke is complete. The v keystroke has a lower execution time than the b (as marked by /) because it is closer to the home key location of the first finger. The earlier completion of the v keystroke has no effect on the initiation of the next, however; instead it produces a longer interval between keys in the v-e digraph than in the b-e digraph, although from initiation to keypress the e keystroke is identical.

```
+ 2.085   PROCEDURAL         FIRED INITIATE-LETTER
  2.085   PROCEDURAL         CLEAR-BUFFER MANUAL
  2.085   PROCEDURAL         CONFLICT-RESOLUTION
/ 2.164   MOTOR              OUTPUT-KEY #(2.75 1.5)
  2.164   [TYPING-WINDOW]    KEY V (289)
  2.164   PROCEDURAL         CONFLICT-RESOLUTION
  2.214   PROCEDURAL         CONFLICT-RESOLUTION
  2.295   PROCEDURAL         CONFLICT-RESOLUTION
+ 2.345   PROCEDURAL         FIRED INITIATE-LETTER
  2.345   PROCEDURAL         CLEAR-BUFFER MANUAL
  2.345   PROCEDURAL         CONFLICT-RESOLUTION
/ 2.424   MOTOR              OUTPUT-KEY #(1.5 0.0)
  2.424   [TYPING-WINDOW]    KEY E (260)

+ 2.085   PROCEDURAL         FIRED INITIATE-LETTER
  2.085   PROCEDURAL         CLEAR-BUFFER MANUAL
  2.085   PROCEDURAL         CONFLICT-RESOLUTION
/ 2.204   MOTOR              OUTPUT-KEY #(3.5 1.5)
  2.204   [TYPING-WINDOW]    KEY B (329)
  2.204   PROCEDURAL         CONFLICT-RESOLUTION
  2.254   PROCEDURAL         CONFLICT-RESOLUTION
  2.295   PROCEDURAL         CONFLICT-RESOLUTION
+ 2.345   PROCEDURAL         FIRED INITIATE-LETTER
  2.345   PROCEDURAL         CLEAR-BUFFER MANUAL
  2.345   PROCEDURAL         CONFLICT-RESOLUTION
/ 2.424   MOTOR              OUTPUT-KEY #(1.5 0.0)
  2.424   [TYPING-WINDOW]    KEY E (220)
```

It is possible to integrate John's analysis into the ACT-R model, even if Rumelhart and Norman's data are for a typist about 25% faster than the ACT-R model. We developed new motor code, modifying execution and finish time computations to match John's analysis. Manual requests were triggered on "state free" rather than "preparation free", and preparation time was zeroed out. These changes allow the model to reproduce TYPIST's performance almost exactly, but they are an awkward fit for ACT-R.

Recall that for a given digraph α-β, the duration of the α_{up} component depends on β. The duration of a keystroke is computed when it is initiated, but when α is initiated, β is not yet available to the motor module. The modified motor code requires manual requests to be made as pairs of keystrokes, current and future, based on the one-key lookahead (in the retrieval buffer) used to handle the end of syllables—an ad hoc solution without theoretical justification. It further proved difficult to accommodate Phenomenon 7 (alternate-hand timing) as a motor phenomenon in the changed code. Incorporating key context into the design of the existing model introduces complexity that we leave for future work.

Phenomenon 12. A concurrent task does not affect typing. Salthouse and Saults (1987) describe an experiment in which typists were asked to type while performing a simultaneous auditory reaction time task. While typing, when the typists heard an auditory cue, they were to press a foot pedal. For this phenomenon, a simple foot pedal motor extension was added to ACT-R. A *pedal* buffer in the motor module might be appropriate, but this was not implemented; instead *press-pedal* requests are interleaved with the hand movements.

Typing performance degraded only slightly for participants. TYPIST and ACT-R were evaluated on the elevator sentence, with an auditory cue beginning at 25 different random locations.

Statistic	Target	TYPIST	ACT-R
Single (ms)	181	195 (7.7%)	210 (16.2%)
Concurrent (ms)	185	196 (5.9%)	212 (14.6%)
Pedal (ms)	431	435 (0.9%)	445 (3.2%)

Phenomenon 13. Copy span is 7–40 characters. The copy span is the number of characters which a typist can continue typing after a single inspection of the material, without its being visible during typing. Salthouse (1986) describes an experiment in which the display was erased after predetermined number of keystrokes by participants, after which they continued typing as much as they remembered. TYPIST and the ACT-R model were evaluated on an equivalent task, in which the elevator sentence was typed and the copy span was determined after each character.

Statistic	Target	TYPIST	ACT-R
Copy span (ms)	14.6	12.5 (14.4%)	15.2 (4.3%)

Phenomenon 14. Stopping span is between one and two characters. The stopping span is the number of characters to which a typist commits to after a signal to stop typing. Logan (1982) describes an experiment in which participants were asked to type single words of 3, 5, or 7 letters; after a predetermined amount of time (500, 650, 800, or 950 ms), an auditory cue was given. TYPIST And ACT-R were evaluated using the same time values, on words that covered all combinations of same- and alternate-hand keystrokes. The results comparing Logan's, TYPIST, and ACT-R model are as follows:

Statistic	Target	TYPIST	ACT-R
Stopping span (char)	1.57	1.76 (12.1%)	2.11 (34.2%)

Overall, the model improves on results obtainable by text entry in CogTool, which types the elevator sentence at approximately 50 wpm. CogTool does not model the differences between words and random letter order, producing mean keystroke intervals for EP, LC, and LJ that differ by at most 5 ms (Phenomenon 5). Alternate-hand keystrokes are 39.5 ms faster than single-hand keystrokes (Phenomenon 7, 12% error). The first keystroke in a word is 17.5% slower than the remaining keystrokes (Phenomenon 10, 12.5% error). CogTool does not model key context (Phenomenon 11).

Discussion

The ACT-R model gives predictions of human performance on basic typing phenomena that are at least qualitatively correct, except for context dependence in keystroke duration and the stopping span being outside an observed boundary.

Motor processing in ACT-R is based on that of EPIC (Kieras & Meyer, 1997), in which the duration of a keystroke depends on several factors: preparation time, motor initiation time, and a minimum burst time, as well as Fitts' Law movement time. By design, the duration of preparation for a movement increases with the number of features that differ from the previous movement (different hands, fingers, direction and distance of movement). Using the default ACT-R keyboard, the press-key movement style, default parameter settings and no motor changes, the model's performance is roughly similar to that described in the previous section. Leaving aside the different typing speed, the differences are on alternate hands (Phenomenon 7, 150% error), first keystroke duration (Phenomenon 10, 161% error), and stopping span (Phenomenon 14, 5.3% error).

We have only lightly explored the space of ACT-R motor parameters, settling on modifications to preparation time as the simplest way to bring modeling results in line with human performance. For the typing model, the feature preparation computations are modified such that preparation of each keystroke has a default duration (50 ms) plus an extra increment when the previous keystroke was with the same hand. Our repurposing of feature preparation for typing is not theoretically well-motivated, in part because theory is sparse. In an updated analysis of the motor literature and EPIC, Kieras (2009) eliminates the dependence of visually aimed manual and ocular movements on feature preparation. He further asks, "Should feature preparation be discarded for keypress movements as well?" For typing the answer appears to be yes, where feature preparation is replaced by functionality that approximates the timing of overlapping, interdependent physical movements with the sequential movements required by the architecture.

Despite the its limitations, we believe this work is important for a few reasons. First is the pragmatic accomplishment of extending ACT-R to a very common task; some human experiments that involve typing as a primary or secondary task can now be taken on. Our work provides evidence for the soundness of TYPIST's design in a symbolic architecture. Second, the performance limitations of the model suggest new directions for research on the architecture, with well-defined tasks and clear empirical targets. Third, the MHP representation makes TYPIST performance easier to analyze than that of the ACT-R model, but we also find value in running trials over large sets of sentences and analyzing aggregate data. Finally, the model raises questions dealing with strategies, visual processing, and how typists learn to adjust their reading speed to working memory limitations, which we will examine in future work.

Acknowledgments

This work was funded by grants from the Army Research Office [W911NF-08-1-0105] and the National Science Foundation [IIS-1451172]. Thanks to Dan Bothell, Bonnie John, Dave Kieras, Don Morrison, and Frank Ritter for code, advice, and suggestions.

References

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

John, B. E. (1996). TYPIST: A theory of performance in skilled typing. *Human-Computer Interaction*, *11*(4), 321–355.

John, B. E., Prevas, K., Salvucci, D. D., & Koedinger, K. (2004). Predictive human performance modeling made easy. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 455–462).

Kieras, D. E. (2009). Why EPIC was wrong about motor feature programming. In *Proceedings of the international conference on cognitive modeling* (pp. 68–73).

Kieras, D. E., & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-computer interaction*, *12*(4), 391–438.

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*(6), 778.

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

Salthouse, T. A. (1984). Effects of age and skill in typing. *Journal of Experimental Psychology: General*, *113*(3), 345.

Salthouse, T. A. (1986). Perceptual, cognitive, and motoric aspects of transcription typing. *Psychological bulletin*, *99*(3), 303.

Salthouse, T. A., & Saults, J. S. (1987). Multiple spans in transcription typing. *Journal of Applied Psychology*, *72*(2), 187.

Snyder, K. M., & Logan, G. D. (2014). The problem of serial order in skilled typing. *Journal of Experimental Psychology: Human Perception and Performance*, *40*(4), 1697–1717.

West, L. J., & Sabban, Y. (1982). Hierarchy of stroking habits at the typewriter. *Journal of Applied Psychology*, *67*(3), 370.

Wu, C., & Liu, Y. (2004). Modeling human transcription typing with queuing network-model human processor (QN-MHP). In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 48, pp. 381–385).