FOCL:
A Probabilistic Character Layout Strategy for Mobile Text Entry

by

Thomas Jeffrey Bellman

A Thesis submitted in conformity with the requirements
for the Degree of Master of Science
Graduate Department of Computer Science
University of Toronto

0-612-40741-1

ABSTRACT

There is an increasingly strong demand on mobile communications devices, such as pagers and cellular phones, to support full-fledged text-entry. In providing text-messaging features, manufacturers have, thus far, failed to consider the unique restrictions of the mobile domain, in particular its limited input and output bandwidth and limited physical space. The result is that interaction is crude and demanding of the user. In one technique, the user moves a cursor around an on-screen character set and selects letters one at a time. This research focuses on improving entry speeds with this technique. By applying statistical data of English to the task of rearranging the character layout after each character entered, we achieved a significantly superior result compared to a fixed layout, after two design iterations. The successful design is one whose layout is partly fixed, and partly fluctuating, combining the advantages of both. It is thus dubbed a *hybrid* layout. The general technique is called the Fluctuating Optimal Character Layout strategy, or FOCL.

ACKNOWLEDGEMENTS

I would like to say thank you to a number of people who have been instrumental in helping me complete this thesis.

- To my thesis advisor, Scott MacKenzie: Thanks for helping me find a topic I could really run with, for always being accessible and reliable, for providing insightful and timely feedback on my ideas and pieces of writing, and for always pushing me to produce something better.
- To my second reader, Mark Chignell: Thanks for providing important insight into the problem, and for reading and critiquing the thesis so promptly during a very busy period.
- To the members of the Input Research Group, in particular, Bill Buxton, Neil Enns, Jay Goldman, Maurice Masliah, Joanna McGrenere, and Jade Rubick: Thanks for your extremely helpful feedback and suggestions on the research as it evolved.
- To John Samuels: Thanks for your much-needed help in managing this large document, and making it look presentable, and for sacrificing sleep to do so.
- To Jamie Bush: Thanks for providing a quiet place to write, and for your frequent words, and beers, of encouragement.
- To Adrienne Rosenwhite: Thanks for showing me how to actually finish. I couldn't have done it without your guidance and—how shall I put it—incentives. I think the thesis would make your mother proud.
- To Jill Lichtblau: Thanks for the wonderful drawings.
- To Mom and Dad: Thanks for raising me to value knowledge and demand high standards of my work, and thanks for all your love and support over the years.
- To Gina and Vivian: Thanks for your love and support, and for getting me to go back to University—not a bad idea!
- To Micah, Amit and Asaf: I'm sorry for all the times this project took me away from you. Whenever the stress of the thesis was getting to me, a look at one of you always comforted me, by reminding me what is truly important in life.
- And finally, to Ruth: What will my excuse be now that "When my thesis is finished" is no longer valid? Now I can *really* do anything you put your mind to. I could never have done *this* without you. There isn't space to thank you adequately for all you have done, and do for me, every day. אני מאד אוהב אותך

TABLE OF CONTENTS

Chapter 1: Overview and Scope

This thesis describes the iterative design and empirical testing of a new text entry method. Although the method may have broader applicability, in the scope of this thesis it is aimed at enabling and/or improving text entry on portable electronic devices. Because the domain of such devices is commonly referred to as mobile computing, we will refer to text entry within this domain as *mobile text entry*.

1.1 Design Requirements and Restrictions

1.1.1 Broad Design Requirements

In developing the method, our goal is to satisfy the following four requirements:

- **Size**

  The method can be implemented on a device that fits in a person's pocket.

- **Feasibility**

  The technology for realizing the design is feasible today.

- **Intuitiveness**

  The interaction technique is intuitive enough to afford competent, though not expert, use in a few minutes.

- **One-Handed Operation**

  The system should support one-handed use.

These requirements serve both as a way of distinguishing the research from other methods of text entry and as one way of evaluating potential design implementations of the method. Many existing text-entry strategies satisfy some of these requirements, but not all four. For example, pen-based input on a palm-top computer is a proven success (Feasibility) and in general requires little time to achieve competence (Intuitiveness), but a palm-top computer is larger than pocket-size (Size), and requires two hands to operate (One-Handed Operation). Speech-driven dictation can be performed without large devices (SIZE), may not require the use of either hand (One-Handed Operation), and is easily learned (Intuitiveness), but has not yet achieved the reliability (Feasibility) required for everyday, let alone mobile, text entry.

1.1.2 Restrictions on Mobile Text Entry Design

The four requirements impose a rigid set of restrictions on the design of a text entry method. This section discusses these restrictions.

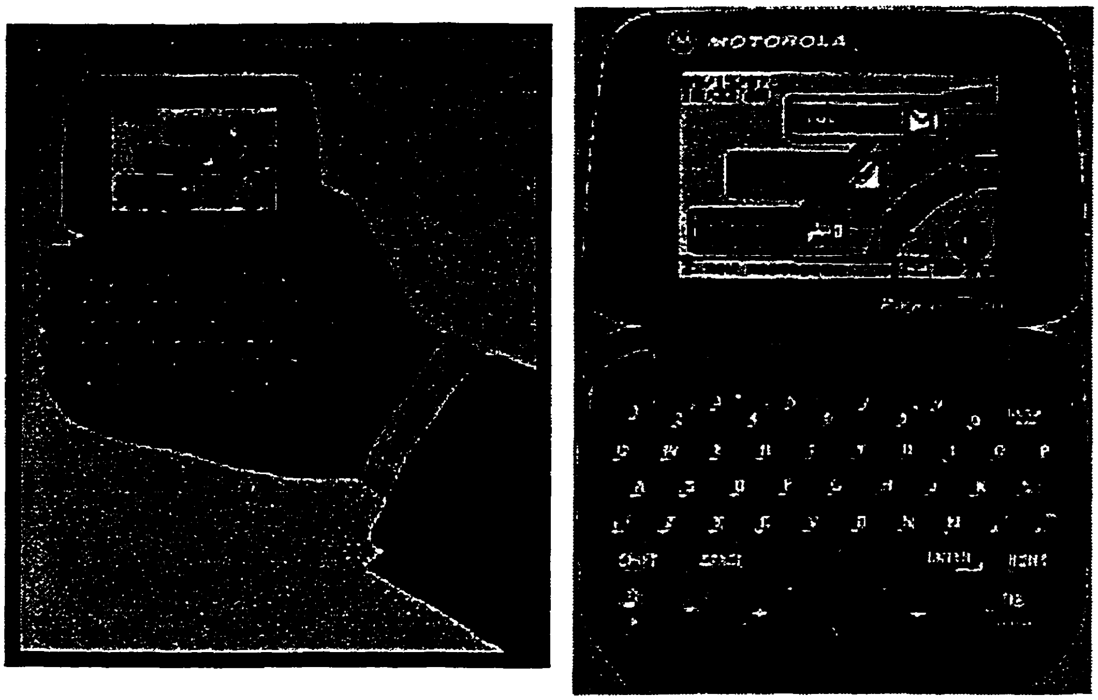


Figure 1-1: Motorola's PageWriter 2000 (from http://www.mot.com/MIMS/MSPG /Products/Two-way/pagewriter)

Although the requirement of one-handed use effectively rules out a full keyboard, the best argument against such an approach is that the small surface area of pocket-sized devices does not truly afford sufficient space for a full keyboard. This has not stopped some companies from taking this approach to mobile text entry. The Motorola *PageWriter 2000*, seen in Figure 1-1, is an example.

While it is true that miniature keyboards are found on a number of portable devices, shrinking keyboards to ever smaller sizes is not a viable way to support mobile text entry. Usability suffers as the size of keys gets smaller (Wilkund & Dumas, 1987). This research is concerned with situations in which a keyboard, even a miniature one, is simply not feasible.

With the keyboard option eliminated, character input must be accomplished either by a small number of buttons or by alternative methods of input. In the absence of a key for every letter, either several characters must be assigned to each key or the character set must be relegated to the device's liquid crystal display (LCD). The assignment of several characters to each key creates a *multifunction keyboard*.[1] Displaying the character set on an LCD is an example of an *on-screen keyboard*. Multifunction keyboards are discussed later, in the context of the telephone keypad, in Section 3.1.1, while on-screen keyboards are discussed in Section 3.1.3. Note that the limited size of typical LCDs on portable devices means that in an on-screen implementation the entire character set may not be visible at once.

In addition to the character set, the LCD must accommodate the output. The display is thus both a character input space and text output space. Although today's displays routinely combine both input and output, the limited space in which a pocket-sized device must accomplish this raises additional design issues. What proportion of the display should be allocated to each function? Should the spatial arrangement of input and output be left/right or top/bottom?

1.1.3 Alternative Interaction Styles Ruled Out

There are a number of interaction techniques with the potential to produce good mobile text-entry solutions that violate one or more of the four requirements on page 1, and thus are *not* the subject of this research.

Speech Recognition and Gesture Recognition

While faster entry speeds may be achievable in one of these areas, the very stringent power, memory size, and processor speed limitations of mobile devices are still an obstacle to realizing solutions using speech and gesture. In the case of speech recognition, such systems are not yet robust enough for mobile situations, where external noise cannot be controlled. These realities will no doubt change in the coming years. In the meantime, this research seeks solutions with more immediate feasibility, in particular, ones that make existing key and button-based techniques more appropriate for the mobile domain.

Pen-Based Input

Excellent pen-based solutions are gaining popularity in palm-top computers. However, the pen-based approach fails to satisfy two requirements of this research. Pen input requires two hands, one to grasp the device and one to write with, thus violating the requirement of one-handed

[1] Although many of the keys on the standard keyboard are multifunction, we tend not to think of it as such.

usability. In addition, it is likely that the required surface area for writing and required volume for storing the stylus necessitate a larger than pocket-sized device.

Chordal Keying

Chord keyboards (Rochester, Bequaert, & Sharp, 1978) have never gained widespread acceptance probably because of the steep learning curve associated with their use (Gopher & Raij, 1988). As such, the chordal approach violates the requirement of intuitiveness.

1.1.4 The Dilemma of Designing for One-Handed Use

Figure 1-2: A reckless vision of mobile computing: anywhere, anytime.

Should we design text-entry methods that can be performed with one hand, given that users may not have both hands free? The pages of *Wired* magazine are sprinkled with images of devices specifically intended for use while driving, riding, or running. While some futurists may envision people performing complex tasks such as text entry while in the midst of another task that *should* demand their full attention (see Figure 1-2), reason and empirical evidence suggest that this vision of technology use is misguided. The National Highway Transportation Safety Association (1997) reports a statistically significant increase in automobile deaths related to cellular phone use as the number of cellular phones in circulation has increased. Cell phone users are disproportionately the drivers of the striking vehicle when involved in a collision (NHTSA, 1997). And a recent article in the New England Journal of Medicine (1997) concluded that the risk of collision is quadrupled during a cellular phone call. It is a vision that emerges, one might argue, from a narrow definition of mobility, namely: *situations in which the body is in motion.*

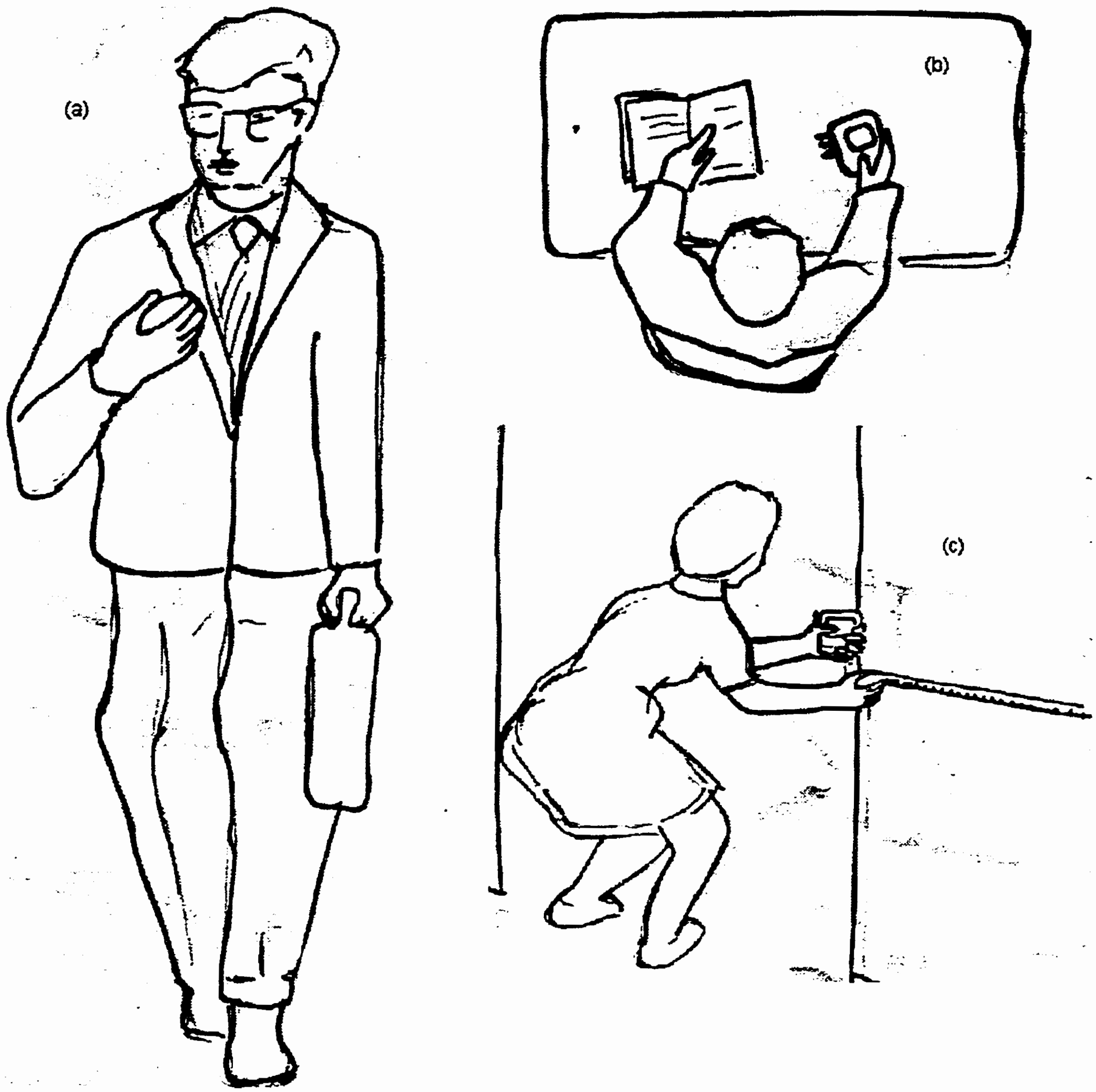


Figure 1-3: One-handed text-entry scenarios

A broader definition of mobility is the following: *situations in which the user cannot count on working indefinitely in their current physical position.* The narrower definition leads to the assumption that one or two of the mobile user's hands are engaged, as in driving or riding a bicycle. According to the broader definition, however, there are mobile situations in which both hands are available, for example on the train or bus, at a restaurant, or in the press box at a sports event.

Still, not all one-handed situations put the user, and others, in peril, so there is clearly a need to support one-handed use. Figure 1-3 illustrates several safe scenarios in which a user has only one hand free for text entry. In mobile situations one hand may be carrying another object that cannot be put down (Figure 1-3a). One-handed text entry might replace handwriting in situations such as taking notes while reading a book or document (Figure 1-3b). Finally, one can imagine tasks that require information to be recorded by one hand while the other hand holds an object steady (Figure 1-3c). Laboratory workers, for example, frequently have to record measurements in this way.

Moreover, one-handed text entry may be beneficial to certain disabled persons lacking the use of both hands. Designing for one-handed use *is*, one can argue, a worthwhile endeavour. Unfortunately, there is little that can be done, with the exception of legislation, to prevent the unsafe use of technology. It is a telling finding that although people perceive the risk of cellular phone use while driving, their awareness is not enough to prevent them from doing it (NHTSA, 1997).

1.2 Scope Limitations

In an effort to complete this research in a reasonable amount of time and within the bounds of the resources available, a number of decisions to restrict the scope of research were made. This section discusses these decisions and their implications.

1.2.1 As-Needed Approach to System-Intelligence

Our approach is bottom-up, introducing as little in the way of system intelligence and complexity as required for reasonable text-entry rates. Thus, our solutions do not use dictionary-based algorithms, for example, to perform word or phrase completion (Darragh, Witten, & James, 1990; Witten & Bell, 1990), even though such techniques hold considerable promise and have already demonstrated their utility with, in particular, physically-challenged users (Darragh, Witten, & James, 1990; Shein, Treviranus, Brownlow, Milner, & Parnes, 1992). This decision is, in part, an attempt to limit the scope of the research. However it also reflects the realization that every new feature requires additional interface components to accommodate it. The decision, thus, preserves the simplicity of the interaction, in keeping with the requirement of intuitiveness (see p. 1).

1.2.2 Non-Alphabetic Characters

In parts of the thesis, particularly in Chapter 4 on the design space, the ability of various text-entry methods--including the method that is the subject of this research--to accommodate non-alphabetic characters is discussed. However, none of the designs developed or tested in this research incorporate characters other than the letters of the alphabet. While this is acknowledged as a limitation, it is felt to be acceptable at this stage of research for the following reasons:

- The greatest proportion of time spent entering text involves the alphabetic characters.
- This research is based largely on frequencies of occurrence of pairs of characters. It is unclear that such data for non-alphabetic characters would be at all meaningful.
- Excluding the digits, punctuation and other symbols from a text-entry system does not severely limit the linguistic expressiveness of the person using it, whereas excluding any or all of the alphabet does.
- Adding ways of dealing with punctuation symbols, digits and other sets of characters at a later stage of research is a manageable design problem.

1.2.3 Letter Case

This research also ignores the issue of letter case. While this defers the problem of switching between cases to a later stage of research, it does not severely limit the expressiveness of the text entry. There was some issue as to whether to use all upper or all lower case in the studies. Upper case was chosen to avoid the appearance of first letters of names and sentences in lower case, which to some is unacceptable usage. In both experiments, all letters appeared in upper case. Again, it is felt that the later addition of a way of dealing with case switching is a manageable design problem.

1.3 Research Question

Thus far, we have stated four broad requirements for the design of a mobile text-entry strategy (see p. 1), and identified the restrictions these requirements impose on a design. Our research question can now be stated as follows:

> *Given these requirements, and the restrictions they imply, can we design a technique for mobile text entry that yields reasonable entry speeds?*

By reasonable we mean entry rates that exceed those of existing text-entry methods for small, input-limited communications devices (e.g., pagers and cellular phones). An additional goal is to match or exceed handwriting speeds, since our method may be considered an alternative to pen-based input. Card, Moran and Newell (1983, p. 61) list handwriting speeds of 732 milliseconds per character (ms/char), which is equivalent to 16.4 words per minute (wpm) using the common assumption of 1 word = 5 characters. Hand printing speeds are reported as a range of 545-952 ms/char, or 12.6 - 22.0 wpm.

1.4 Proposed Solution

The method of text entry developed to address this research question is called the *Fluctuating Optimal Character Layout* strategy, or FOCL. FOCL is an extension of the on-screen keyboard interaction style, in which the character set is displayed on screen. Though commonly used with stylus tapping, in this case arrow keys on the device move a cursor around the layout. In the FOCL technique, the character layout is rearranged optimally *after each character entered*, so as to reduce the number of keystrokes to the next character. While there are many ways to optimize a layout of characters, in the current implementation of FOCL the layout is optimized according to the relative probabilities of digrams[2] in common English. The general idea, however, refers to any implementation in which the character layout is frequently rearranged, or fluctuated, during use, with the intent of increasing entry rates.

Figure 1-4 shows a device for which FOCL would be suitable. The *AccessLink*™ two-way pager, by Wireless Access Corp. (Santa Clara, CA), supports full text entry. It has a 4-line, 20-character alphanumeric display, and six buttons for input.

---

[2] Also referred to as digraphs, bigraphs and bigrams in the literature. The terms are synonmous, meaning two-letter sequences. *N*-gram refers to a sequence of *N* characters.

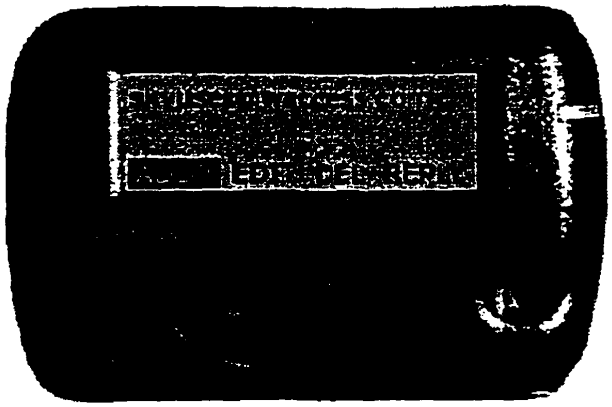


Figure 1-4: The AccessLink™ two-way pager by Wireless Access, with text entry capability. (from http://www.wirelessaccess.com)

In text-entry mode, the top two lines of the display are for output, the bottom two for displaying the character set. While the character set occupies four lines, only two are viewed at a time. The entire character set is shown in

Figure 1-5.

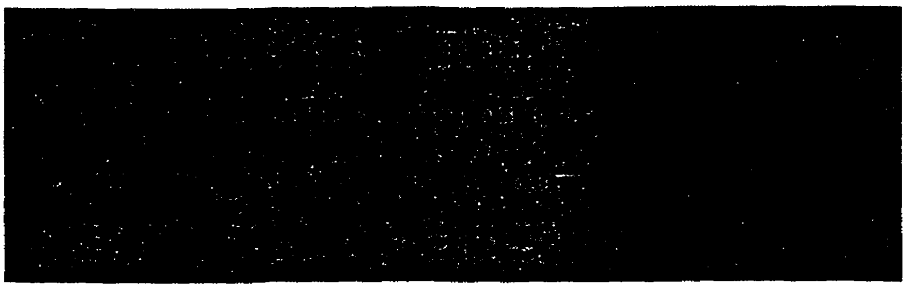


Figure 1-5: Character layout for text entry on the AccessLink™ two-way pager.

The button at the centre of the device functions like a four-directional joystick. Pressing the left or right sides of the button moves a cursor horizontally on the display. Pressing the upper or lower sides moves the cursor vertically or, depending on the current vertical location of the cursor and direction of key press, scrolls the display. The centre of the button performs selection, causing the character at the cursor position to appear in the output section of the display. To enter text, users repeatedly move the cursor to a desired character and select it.

Clearly, this is a crude way of inputting text. This research addresses the need for faster and less demanding text entry on devices of this sort. The emergence of products such as in Figure 1-4, as well as the recent slew of digital cellular services offering text-messaging features, shows that a demand for good text entry on portable communications devices exists. However, unless researchers can simplify message composition, the vast majority of users will reject this capability, as was the case with VCR programming.

1.5 Outline of Thesis

The remainder of this thesis consists of six chapters. Chapter 2 outlines the three models underlying this work: a linguistic-information theory model; a keystroke-level model for making a priori evaluations of potential designs; and a learning model that uses empirical results to predict future performance. In 0 the various aspects of the design space are mapped out, followed by a description of the first experimental prototype, and the rationale behind its design. Chapter 4 describes the structure, execution, and results of a longitudinal experiment to test the prototype. Chapter 5 includes a discussion of redesign considerations motivated by the experimental results, and the design rationale for the consequent second prototype. After describing modifications to the experimental design, Chapter 6 reports the results of a second experiment to test the redesigned prototype. Finally, in Chapter 7, a summary of the research is followed by a discussion of the future potential of the FOCL technique and possible paths of further investigation.

Chapter 2: Models

2.1 Introduction

Like many research projects in human-computer interaction, this one began with a software concept. The idea's simplicity and novelty aroused curiosity, and as a result, before any attempt to justify it theoretically, or to seek examples of previous related work, a prototype was rapidly built. Much of the theoretical work described in this chapter came subsequently.

This chapter describes three theoretical models that have informed the development of the FOCL technique. The first is a linguistic-information theory model, whose relevance to this work was discovered only after an experiment was carried out on the first prototype. In particular, it was found that Claude Shannon's (1948, 1951) visions of machine message decoders described behaviour very similar to that desired for a FOCL system. The second model is a keystroke-level model (Card, Moran & Newell, 1983), used to compute the keystroke-to-character ratio given a description of a FOCL design. The keystroke-level model serves as a tool for comparatively evaluating competing designs. The third model is a learning model, used to predict future performance beyond the scope of empirical studies. This model is based on the power law of practice (Crossman, 1959; de Jong, 1957; Newell & Rosenbloom, 1981).

2.2 Linguistic-Information Theory Model

2.2.1 The Statistical Structure of Language

A language is, by its very nature, not random, but highly constrained. Only a small proportion of all the possible letter sequences constitute meaningful words, and only a small proportion of all possible word sequences constitute grammatical sentences. Moreover, the frequencies of occurrence of letter sequences and words vary dramatically. In a diverse, 100,000 word sample of English (Dewey, 1923), the nine most common words accounted for over 25% of the sample, the 69 most common for over 50%, and the 732 most common for over 75%. Similarly, in a sample of English text consisting of 5000 digrams (Friedman, 1960), only 432 or 64% of a possible 26^2 (676) distinct digrams occurred, just 18 distinct digrams accounted for 25% of the

5000, 53 accounted for 50%, and 117 for 75%. And in a 50,000 letter sample of U.S. Government plain-text telegrams (Friedman, 1960), only 56 trigrams (3-letter sequences) and 54 tetragrams (4-letter sequences) occurred more than 100 times.

The non-randomness of language makes it possible to decipher coded messages. Consider an anagram, which encodes a message by replacing all occurrences of a letter with another letter. Even at the outset, before any of the puzzle has been solved, there are constraints on the possible mappings of certain positions in the message. For instance, first letters of words are not likely to be Z or X. The most frequent letters are likely to be one of the most common letters in English, such as E, T, N, and S. There is a good chance that 3-letter words occurring more than once in the message are THE or AND. As a speaker of the language, the decoder has an intuitive knowledge of the rules of its grammar and its statistical structure vis-a-vis the likelihood of different letter sequences. As more of the message is revealed the degree of uncertainty surrounding the identity of particular positions in the coded message is reduced. For example, in Figure 2-1a, with only the C in position 3 known, there are a number of good candidates for the fourth position of the word, including H, T, and E. In Figure 2-1b, however, only E remains a viable choice.

(a)	_ _ C _	TACT?	(b)	O N C _	ONCE!
		SUCH?			
	S R X T	RICE?		S R X T	

Figure 2-1: Anagram chunk at two stages of solution, illustrating how knowledge of surrounding context constrains possibilities. In (a) there are several candidates for the fourth position, while in (b) there is only one.

On *Wheel of Fortune*, the popular television game show based on Hangman, players try to determine a phrase given only blank slots where the words should be. Thus all they know initially is the number of words in the phrase and the number of letters in each word. With each turn, contestants ask the host whether a certain letter is present in the phrase, and if it is, all occurrences of that letter are revealed. The guesses are, of course, not random, but rather demonstrate an intuitive knowledge of the non-random structure of English. The first guess in a round will typically be a letter with one of the highest frequencies. As more of the letters are revealed, guesses reflect the decreasing uncertainty surrounding particular letter positions in the phrase. Code deciphering would be impossible, and *Wheel of Fortune* tedious and boring, if language were random, or if human beings lacked intuitive knowledge of the structure of language. The same type of knowledge is more developed in anagram puzzle buffs, allowing them to make strong guesses and thus break codes quickly.

2.2.2 The Work of Claude Shannon

In his mathematical theory of communication, Shannon (1948) formalized these concepts by introducing two related statistical measures, *entropy* and *redundancy,* for describing the predictability of a language. Entropy is defined as the amount of information that is produced, on average, for every letter of text in the language, and is measured in bits per letter. Redundancy, expressed as a percentage, is defined as "the amount of constraint imposed on a text in the language due to its statistical structure, e.g., in English the high frequency of the letter E, the strong tendency of H to follow T or of U to follow Q." (Shannon, 1948, p. 50). If a language were totally predictable its redundancy would be 100%. Shannon estimated that English may be as much as 75% redundant.

There are orders of entropy and redundancy, corresponding to the range of the statistics of the language taken into account. In general, *N*-order entropy measures the average predictability of the next letter, given that we know the preceding *N-1* letters. Intuitively, the more previous letters we know, the fewer possibilities remain for the next letter--as Figure 2-1 illustrates--and thus the fewer bits needed to represent the set of possibilities. The ability of native speakers to predict English improves substantially with knowledge of up to eight to ten preceding characters (Darragh, Witten, & James, 1990).

Shannon performed numerous experiments which demonstrate both the implicit knowledge speakers of a language possess of its statistical structure and by extension, the potential of machine systems to generate meaningful language when equipped with statistics of the language and the cumulative content of a message. In one experiment (1951, p. 54) aimed at estimating the extent to which English is predictable, subjects were given a series of *Wheel of Fortune*-like blank phrases--in this case with no indication of where spaces occurred--and asked to guess each letter in turn. If the guess was wrong, the subject was told the correct letter. Whatever the outcome of the guess, the subject would record the correct letter. Figure 2-2 shows some actual output from this experiment, with the correct phrase recorded by the subject and a second line in which dashes represent correct guesses and letters represent incorrect guesses.

In the example shown, the subject guessed 69% of the letters correctly. Overall data are not reported, but it is reasonable to assume that this is a typical example. Note that the phrases in Figure 2-2 are parts of larger grammatical sentences, but are not themselves grammatical. This prevented subjects from relying on the rules of sentence structure to guess, which would have yielded an even higher percentage of correct guesses.

```
(1) THE ROOM WAS NOT VERY LIGHT A SMALL OBLONG

(2) ----ROO------NOT-V-----I------SM----OBL----

(1) READING LAMP ON THE DESK SHED GLOW ON

(2) REA----------O------D----SHED-GLO--O--

(1) POLISHED WOOD BUT LESS ON THE SHABBY RED CARPET

(2) P-L-S-----O---BU--L-S--O------SH-----RE--C------
```

Figure 2-2: Sample phrases and output from experiment to measure the predictability of English. (from Shannon, 1951)

In a variation of this experiment, subjects were asked to guess each letter repeatedly, until they guessed correctly (1951, p.56). In this study, the number of guesses was recorded below each letter, as shown in Figure 2-3. In the example, 79 or roughly 80% of 102 characters were guessed correctly on the first guess, 8 on the second guess, 3 on the third guess and 2 on each of the fourth and fifth guesses. Only 8 characters required more than 5 guesses.

```
(1) THERE IS NO REVERSE ON A MOTORCYCLE A
(2) 111511211211151171112132122711114111113
(1) FRIEND OF MINE FOUND THIS OUT
(2) 8613111111111111621111111211111
(1) RATHER DRAMATICALLY THE OTHER DAY
(2) 4111111115111111111111161111111111
```

Figure 2-3: Sample output from experiment recording number of guesses to correctly guess each letter of a string of text. (from Shannon, 1951)

These experiments provide a good metaphor for the desired behaviour of a FOCL-based text entry system, where the FOCL user is the experimenter and the FOCL system the subject in Shannon's experiments. The message a user composes is "revealed" to the system one character at a time, at which point the system "guesses" the next character by placing its guess at the cursor position. The second guess is placed in the closest position to the cursor, the third guess in the next closest position, and so on. The hope is that the next letter in the message will not be too far away from the cursor. The *Reactive Keyboard* (Darragh, Witten, & James, 1990) takes this approach to a much higher statistical order in an attempt to speed up two-handed typing. Using a combination of *N*-gram statistics and adaptive modelling[3], it makes long predictions of potential future text from which the user may optionally select.

[3] A technique that uses the text already input, as opposed to a fixed corpus, as a basis for its predictions.

Table 2-1: Shannon's series of approximations of English using progressively higher-order statistics. (from Shannon, 1951)

1.	Zero-order approximation (symbols independent and equiprobable) XFOML RXKHRJFFJUJ ZLPWCFWJCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZLHJQD
2.	First-order approximation (symbols independent but with frequencies of English text) OCRO HLI RGWRR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL
3.	Second-order approximation (digram structure as in English) ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE
4.	Third-order approximation IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE
5.	First-order word approximation. Rather than continue with tetragram, ..., *N*-gram structure it is easier and better to jump at this point to word units. Here words are chosen independently but with their appropriate frequencies. REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD THESE
6.	Second-order word approximation. The word transition probabilities are correct but no further structure is included THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVE TOLD THE PROBLEM FOR AN UNEXPECTED.

Shannon states that good prediction does not require knowledge of more than *N* preceding letters of text, where *N* is "fairly small", (1951, p.55). This suggests that a FOCL system with "knowledge" of the relative frequencies of *N*-grams will place the target letter close to the cursor a large proportion of the time, and thus produce good entry speeds. Obviously, predictions will improve as *N* increases. To illustrate, consider the word QUIT. U will be the system's first guess after Q is entered regardless of the size of *N*. However, if $N = 2$, then the system's guesses after U is entered will reflect the relative probabilities of digrams of the form U-letter, which would place the letter I at quite a distance from the cursor due to the low probability of the digram UI. If, on the other hand, $N = 3$, then I is certain to be among the first few guesses after U is entered, due to the strong likelihood of a vowel to follow the digram QU. While increasing *N* improves the accuracy of predictions, it also carries the disadvantage of exponentially increasing the size of the lookup table used to "guess".

Shannon (1948) demonstrated mathematically-generated sequences that increasingly approximate English by using progressively higher order *N*-gram statistics. The way in which the results, shown in Table 2-1, become increasingly more coherent, is striking. Witten and Bell (1990) performed a similar process with equally compelling results.[4]

2.2.3 Optimal Keyboard Layouts

Numerous researchers have devised optimal keyboard layouts for a variety of purposes and applications, for example stylus-tapping (Ichbiah, 1992, Zhang, 1998), typing in a language other than English (Marmaras & Lyritzis, 1993), and for the physically challenged (Shein, et al., 1992). These efforts attempt to increase entry rates by exploiting the statistical structure of language to improve the efficiency of text entry.

Optimal layout approaches typically apply first order entropy (single letter frequencies) to determine optimal positioning of letters in the layout, so that the most common letters are the most accessible. In many cases, second order entropy (digram frequencies) is also employed, so that common letter pairs can be entered as efficiently as possible. Efficiency has different design implications for different interaction styles. For example, in stylus tapping the goal is to minimize stylus motion. Therefore a good design increases the proximity of frequently co-occurring letters. In two-handed typing the goal is to reduce difficult finger combinations. Therefore a good design places each of the two letters of common digrams under a different hand.[5]

Due to the large number of digrams in English--or any language--it is difficult to take all letters into account in a fixed layout design. Often only the most common ones impact the design (e.g., Marmaras & Lyritzis, 1993, p. 292). It is unrealistic to employ third or higher order entropy in the design of a fixed optimal layout--a bit like evaluating every possible chess move before moving.

---

[4] Witten and Bell performed their approximations with the aid of computers, while Shannon had to devise ways—rather brilliant ones, at that--of doing it manually. Shannon was forced to move to the level of words for fourth order approximation due to exponential growth in the work involved. It turns out that this is not a simplification of language, as the results of Witten and Bell, who did not move to the word level, show. From the 5th order approximation, the output contains words exclusively, i.e., no meaningless strings.

[5] Despite widespread claims to the contrary, the QWERTY keyboard layout largely satisfies this criterion, though for a reason other than efficiency. To prevent fast typists from jamming the keys of the mechanical typewriters used at the time, it was necessary to space common letter pairs as much as possible, which often meant placing the two letters under opposite hands. While it is true that the QWERTY layout was designed for two-fingered typing (Potosnak, 1988), its viability for two-handed typing is well

2.2.4 Fluctuating Optimal Character Layout (FOCL)

By adding a temporal aspect to keyboard layout, FOCL provides a realistic opportunity to explore the effects of higher order entropy on the design of a text entry method. In the FOCL technique the layout is optimized continually--after each character entered. In the current FOCL implementation, each new layout is only *second order optimal*, that is, optimal with respect to digram frequencies. While this is the logical starting point given the goal of adding complexity only as needed (see p. 6), there is no reason to rule out a FOCL implementation where $N > 2$, i.e., one that uses the probabilities associated with the last three or more letters entered.

2.2.5 N-Gram Data Sources and Variability

The sources of *N*-gram frequency data are numerous and varied. Cryptanalysts, or code breakers, use these data extensively (e.g., Friedman, 1960; Pratt, 1942). Researchers in the psychology of reading and learning (e.g., Underwood & Schulz, 1960) have an interest in *N*-grams and how children come to recognize certain ones, but not others, as units. They are of interest to linguists as one way of describing a language (e.g., Mayzner & Tresselt, 1965). Finally, ergonomists, industrial engineers and HCI researchers (Duncan & Ferguson, 1974; Marmaras & Lyritzis, 1993; Maxwell, 1935) use *N*-gram data as a basis for producing intelligent keyboard layouts for specific or general applications.

In implementing a FOCL system, the choice of *N*-gram probability data s critical, as it has a direct impact on the system's effectiveness. If the sample used to generate the data is not representative of the text that is being entered, then the system is not likely to be very good. Issues surrounding the choice of *N*-gram data are discussed in this section.

Stylistic Variability

It is impossible to know the exact probabilities of *N*-grams in a language, since any estimate is based on a finite corpus of text. Confidence in estimates of *N*-gram probabilities is strengthened as they are shown to correlate highly across different corpora (Mayzner & Tresselt, 1965). Stylistically different samples of English differ from one another in their statistical structure. For example, a corpus consisting exclusively of journalistic pieces will yield rather different *N*-gram data from one made up of classic literature, or informal correspondence. Figure 2-4 shows alphabetic orderings in decreasing order of single letter frequency for samples of literary and

---

demonstrated, and claims of its inferior performance are highly exaggerated (Norman & Fisher, 1982 ; Potosnak, 1988) telegraphic English (Friedman, 1960). Even the statistical structure of seemingly similar English samples can vary greatly (Darragh, et al., 1990).

Literary:	E T O A N I R S H D L U C M P F Y W G B V K J X Z Q
Telegraphic:	E O A N I R S T D L H U C M P Y F G W B V K X J Q Z

Figure 2-4: Alphabetic orderings by decreasing single letter frequencies for 2 stylistically different samples of English text. (from Friedman, 1960)

To capture accurate N-gram probabilities of a language *as a whole,* researchers have assembled corpora from many stylistically different samples of text. The corpora used in Underwood & Schulz (1960) and Mayzner & Tresselt (1965) consisted of 100 200-word samples from a variety of sources, including magazines, newspapers, fiction, and non-fiction. Starting points within a given source were chosen randomly.

Historical Variability

Languages change over time. As words and idioms enter and leave common usage, and acceptable writing styles evolve, the relative N-gram probabilities of English change. To maintain optimality therefore requires periodic adjustment. However, users tend to resist change that requires new learning (Anstey, 1988; Zipf, 1949). For example, typists have not flocked to the Dvorak keyboard (Dvorak, 1943). despite its alleged superiority (Norman & Fisher, 1982). For widespread acceptance, a standardized layout is required, even though over time it may cease to be the best one available.

Intra- and Inter-Linguistic Variability

In addition to style, dialect also has an effect on the statistical structure of a language. British, American, Canadian, East Indian and West Indian English are only some of the dialects of English, each with a different N-gram distribution. And while the FOCL strategy can be applied to any language, speakers of two languages with similar scripts (e.g., English and French) might be confused if the system is not as efficient in one language because its predictions reflect the statistical structure of the other.

Cross-Disciplinary Data Incompatibility

Different areas of inquiry have differing demands on the data. Data prepared to meet the needs of one discipline may be incomplete for the purposes of another. For example, the data of the aforementioned studies do not include the space character, which, at 18%, is the most commonly occurring character, and thus of extreme importance to text-entry design. Noting that the Mayzner & Tresselt data (1965) recorded position within a word, Soukoreff and MacKenzie

(1995) inferred digrams of the type SPACE-LETTER (beginning of word) and LETTER-SPACE (end of word), and thereby generated digram frequency data appropriate to text entry design by its inclusion of the space character. Their data are shown in Table 2-3 on p. 22.

Another feature of *N*-gram data that can lead to cross-disciplinary incompatibilities is the size of words included in the corpus. For example, Mayzner & Tresselt (1965) only included words from three to seven letters in length in their sample. Their data may not be appropriate as a basis for creating optimal keyboard layouts, since one- and two- letter words are very common in text composition. Letters that appear disproportionately in the omitted words will be under-represented in the sample. A common complaint of subjects in the first experiment, which used the Soukoreff and Mackenzie data, was the large distance of the letter I from the cursor in the layout following the entry of a space character. This problem was rectified in experiment 2 by using data based on a corpus that included words of any length. The inclusion of one- and two-letter words raised the frequency of the digram SPACE-I, causing the letter I to be positioned closer to the cursor after a space.

2.2.6 Choosing Appropriate *N*-Gram Data

An important consideration in choosing appropriate *N*-gram probability data for a FOCL implementation is whether the system will be used to express an array of styles, or one particular style. If the stylistic scope of the system is narrow, then data appropriate to the style in question should be sought. If no appropriate data are available, there are two options: choosing an existing data set--if that is an acceptable compromise--or generating new data. If an existing data set is incomplete in some way, one may decide to enhance it (Soukoreff & MacKenzie, 1995).

2.2.7 Assembling Original *N*-Gram Data

Before the ubiquity of the personal computer, tabulating *N*-gram frequency data was time-consuming and required large staffs of research assistants, and/or the considerable use of mainframe computer time. Since most of the published tables date from this time, it is not surprising that these data are based on samples of no more than 100,000, and more commonly 20,000, words.

The ease with which such data are now assembled with personal computers suggests that rather than use or enhance existing data sets, researchers ought to generate their own data, based on the much larger samples that today's computers make practical. After determining that all the available data sets were in some way inappropriate to the FOCL problem, a large corpus was assembled to generate more appropriate digram probabilities.

Project Guttenberg (http://www.promo.net/pg/) is a non-profit effort to assemble as much of English literature online as possible, and make it available to anyone with a computer and a modem. Due to copyright concerns, submissions (books are scanned or typed on a volunteer basis by anyone who cares to do so) tend not to be current, although there are numerous exceptions. The collection includes both fiction and non-fiction books.

Table 2-2: Books used in original corpus assembled to generate *N*-gram data appropriate to mobile text entry.

Title and Author	Words	Type of book	First published
Anne of Green Gables by: Lucy Maud Montgomery	102,816	Fiction	1908
The Black Experience in America by: Norman Coombs	76,681	Non-fiction	1972
A Brief History of the Internet by: Michael S. Hart	16,396	Non-fiction	1995
The Mysterious Affair at Styles by: Agatha Christie	57,009	Fiction - Mystery	1920
Secret Agent by: Joseph Conrad	91,212	Fiction	1907
Tom Swift and his Air Glider by: Victor Appleton	41,664	Children's fiction	1951
CORPUS SIZE:	385,778		

Six books were selected for the corpus. The titles, authors, word lengths, type of book and date of first publication are shown in Table 2-2. The selection of these particular books represents an effort to capture a variety of styles of English. Two of the books are non-fiction. The books span nearly a century, ranging from Joseph Conrad's Secret Agent, published in 1907, to Project Guttenberg founder Michael Hart's A Brief History of the Internet, published in 1995. The authors include both males and females. Some of the books are serious, others are less so. The books cover a range of reading difficulties as well; the Tom Swift book is written in language accessible to pre-teens and teenagers, while Joseph Conrad's writing is amongst the most sophisticated in English literature.

This approach to assembling a corpus is markedly different from that of Mayzner & Tresselt (1965) and Underwood & Schulz (1960), who selected small samples from a large number of sources. It is felt that the much larger size of the corpus, nearly 400,000 words vs. 20,000, offsets any sampling bias.

Text Preparation

The texts were prepared for processing in a number of ways. First, disclaimers, introductions and such, added by the Project Guttenberg, were stripped. All chapter and section headings (e.g., Chapter 1, Part 3) were removed, though not their titles (e.g. Poirot's Solution). The entire text was then converted to upper case, to simplify the requirements for the counting program. A blank character was added before each new paragraph, so that the first letter of the paragraph would be included in a digram of the form SPACE-LETTER.

Digram-Counting Program

A program was authored to count digrams in the corpus. Although it was used to count digrams exclusively, the program is generic and could handle *N*-grams of any length. The program counts digrams whose two constituent characters are in the FOCL character set. When another character is encountered, the variable storing the current digram is flushed and counting only resumes when the next two FOCL-legal characters are read.

2.2.8 Digram Frequency Data

Table 2-3 contains the digram frequencies from Soukoreff & Mackenzie (1995). These data were used to determine the alphabetic sequences that generated FOCL layouts in the first experiment, which are shown in Table 2-4. Each row of Table 2-4 is a sorting of the alphabet in order of likelihood to follow its row heading, assuming

Table 2-3 is representative of common English. So, for example, the most likely letter to follow A is N, then R, then T, and so on. The space character, indicated by the underscore character in Table 2-4, has its own row, representing the order of likelihood to begin a word. A different set of alphabetic sequences, generated from the digram frequencies of the corpus described in Table 2-2, and shown in Table 2-5, was used to generate layouts in the second experiment prototype.

Table 2-3: Digram frequencies from Soukoreff and Mackenzie (1995), modified from Mayzner & Tresselt (1965) to include the space character.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	SPACE	TOTAL
A	2	144	308	382	1	67	138	9	322	7	146	664	177	1576	1	100	0	802	683	785	87	233	57	14	319	12	50	7086
B	136	14	0	0	415	0	0	0	78	18	0	98	1	0	240	0	0	88	15	7	256	1	1	0	13	0	36	1417
C	368	0	13	0	285	0	0	412	67	0	178	108	0	1	298	0	1	71	7	154	34	0	0	0	9	0	47	2053
D	106	1	0	37	375	3	19	0	148	1		22	1	2	137	0	0	83	95	3	52	5	2	0	51	0	2627	3770
E	670	8	181	767	470	103	46	15	127	1	35	332	187	799	44	90	9	1314	630	316	8	172	106	87	189	2	4904	11612
F	145	0	0	0	154	86	0	0	205	0	0	69	3		429	0	0	188	4	102	62	0	0	0	4	0	110	1561
G	94	1	0	0	289	0	19	288	96	0	0	55	1	31	135	0	0	98	42	6	57	0	1	0	2	0	686	1901
H	1164	0	0	0	3155	0	0	1	824	0	0	5	1		487	2		91	8	165	75	0	8	0	32	0	715	6733
I	23	7	304	260	189	56	233	0	1	0	86	324	255	1110	88	42	2	272	484	558	5	165	0	15	0	18	4	4501
J	2	0	0	0	31	0	0	0	9	0	0	0	0	0	41	0	0	0	0	0	56	0	0	0	0	0	0	139
K	2	0	0	0	337	0	0	0	127	0	0	10	1	82	3	1	0	0	50	0	3	0	0	0	8	0	309	933
L	332	4	6	289	591	59	7	0	390	0	38	546	30	1	344	34	0	11	121	74	81	17	19	0	276	0	630	3900
M	394	50	0	0	530	6	0	0	165	0	0	4	28	4	289	77	0	0	53	2	85	0	0	0	19	0	454	2160
N	100	2	98	1213	512	5	771	5	135	8	63	80	0	54	349	0	3	2	148	378	49	3	2	2	115	0	1152	5249
O	65	67	61	119	34	80	9	1	88	3	123	218	417	598	336	138	0	812	195	415	1115	136	398	2	47	5	294	5776
P	142	0	1	0	280	1	0	24	97	0	0	169	0	0	149	64	0	110	48	40	68	0	3	0	14	0	127	1337
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	66	0	0	0	0	0	0	66
R	289	10	22	133	1139	13	59	21	309	0	53	71	65	106	504	9	0	69	318	190	89	22	5	0	145	0	1483	5124
S	196	9	47		626	0	1	328	214	0	57	48	31	16	213	107	8	0	168	754	175	0	32	0	34	0	2228	5292
T	259	2	31	1	583	1	2	3774	252	0	0	75	1	2	331	0	0	187	209	154	132	0	84	0	121	1	2343	8545
U	45	53	114	48	71	10	148	0	65	0	0	247	87	278	3	49	1	402	299	492	0	0	0	1	7	3	255	2678
V	27	0	0	0	683	0	0	0	109	0	0	0	0	0	33	0	0	0	0	0	1	0	0	0	11	0	0	864
W	595	3	0	6	285	0	0	472	374	0	1	12	0	103	264	0	0	35	21	4	2	0	0	0	0	0	326	2503
X	17	0	9	0	9	0	0	0	10	0	0	0	0	0	1	22	0	0	0	23	8	0	0	0	0	0	21	120
Y	11	10	0	0	152	0	1	1	32	0	0	7	1	0	339	16	0	0	81	2	1	0	2	0	0	0	1171	1827
Z	3	0	0	0	26	0	0	0	2	0	0	4	0	0	2	0	0	0	3	0	0	0	0	0	3	9	2	54
SPACE	1882	1033	864	515	423	1059	453	1388	237	93	152	717	876	478	721	588	42	494	1596	3912	134	116	1787	0	436	2	0	19998
TOTAL	7069	1418	2059	3770	11645	1549	1906	6739	4483	131	932	3885	2163	5241	5781	1339	66	5129	5278	8536	2701	870	2507	121	1855	52	19974	107199

Table 2-4: Alphabetic sequences sorted in order of likelihood to follow each letter, based on digram frequency data from Table 2-1.

A	NRTSLDIYCVMKBGPUFW_XZHJAEOQ
B	EUOALRI_JSBYTMVWCDFGHKNPQXZ
C	HAOEKTLRI_UCYSNQBDFGJMPVWXZ
D	_EIOASRUYDLGVFTNWBJMCHKPQXZ
E	_RNDASELTYMCVIWFPXGOKHQBUZJ
F	OIREA_TFLUSYMBCDGHJKNPQVWXZ
G	_EHORIAULSNGTYBMWCDFJKPQVXZ
H	EAI_OTRUYSWLPHMBCDFGJKNQVXZ
I	NTSLCRDMGEVOKFPAZXBU_QIHJWY
J	UOEIABCDFGHJKLMNPQRSTVWXYZ_
K	E_INSLYOUAMPBCDFGHJKQRTVWXZ
L	_ELIOADYSUTFKPMWVRGCBNHJQXZ
M	E_AOIUPSBMYFLNTCDGHJKQRVWXZ
N	D_GETOSIYACLKNUJFHQVBRWXMPZ
O	URNMTWO_LSPVKDIFBACYEGZJXHQ
P	ELOA_RIUPSTHYWCFBDGJKMNQVXZ
Q	UABCDEFGHIJKLMNOPQRSTVWXYZ_
R	_EOSIATYDNULRMGKCVHFBPWJQXZ
S	_TEHIOAUSPKLCYWMNBQGDFJRVXZ
T	H_EOAISRTUYWLCBGNDFMZJKPQVX
U	TRSN_LGCMEIBPDAFYOZQXHJKUVW
V	EIOAYUBCDFGHJKLMNPQRSTVWXZ_
W	AHI_EONRSLDTBUKCFGJMPQVWXYZ
X	TP_AICEUOBDFGHJKLMNQRSVWXYZ
Y	_OESIPABLTWGHMUCDFJKNQRVXYZ
Z	EZLASYIO_BCDFGHJKMNPQRTUVWX
_	TAWSHFBMCOLPDRNGYEIKUVJQZX_

Table 2-5: Alphabetic sequences sorted in order of likelihood to follow each letter, based on digram frequencies of original corpus (Table 2-2).

A	NTSRLDICVMYBGPKFUWZEHJOXAQ
B	ELUOAYRISTBJVMDHNCFWGXKPQZ
C	OEHATIKRLUCYSQNZPBDVFGJMWX
D	EIOARSUYLDNGMVFWTJHBKCPQXZ
E	RDNSALTECMVYXIPFWGOHQUKBJZ
F	OREIAFUTLSYWNDPVHBCGJKMQXZ
G	HEROAILUSNGTYMDFWBVCJKPQXZ
H	EAIOTURYNLSBMFWDCVHQGJKPXZ
I	NTSCOLDERMGAFVPBKZXUQHIJYW
J	UOEAIRPBCDFGHJKLMNQSTVWXYZ
K	EINSYLAFUOWGMBCDHTRJKPQVXZ
L	ELIAYODTUFSKPMVWRBCNGHJQXZ
M	EAIORPUYMBSNFLTCDHVWGJKQXZ
N	DGETOSCIANYKLFUVMWHJQBRXPZ
O	NURFMTWOLSPVCIDKBAHGEYXJZQ
P	EORALPIUTSHYMBWNCFGKDVZJQX
Q	UABCDEFGHIJKLMNOPQRSTVWXYZ
R	EOIASTYDLNRUMCKGPVFHWBJQXZ
S	TEHIOSAUPLCKMNYWBFQDGRVJXZ
T	HOEIARTSUYLWCMFNBGPDVZJXKQ
U	TSRLNGPCEMIADBFOYXKZVHQUJW
V	EIAOLYUMRSYCBDFGHJKNPQTWXZ
W	AHEIONSRLFDYTBUKCMWPGJQVXZ
X	PTXCIAEOHUQYBDFGJKLMNRSVWZ
Y	OESITNALMBCPWRHUFDGZJKQVXY
Z	EAIZYLOUGTWBCDFHJKMNPQRSVX
_	TAISWHOMBCFDPLNREGYUVKJQZX

2.3 Keystroke-Level Model

2.3.1 Task Analysis

The task of entering a letter with the FOCL method is a three step process:

1. Locate the letter by visual search, or anticipate its location
2. Move the cursor to the letter
3. Press the selection key

While other researchers have built simple mathematical models to predict performance on a variety of interaction styles with moderate to good success (Soukoreff & MacKenzie, 1995; MacKenzie, Zhang & Soukoreff, 1998), attempts to do so for FOCL yielded predictions that did not match empirical data. This is probably due to a poor understanding of the dynamics of visual search within the FOCL task (step 1). Instead, our approach is to focus on step 2, that is, to find designs that minimize the distance between letters, thereby reducing keystrokes.

2.3.2 Mean Keystrokes per Character

During the design process, possible FOCL designs were compared by their keystroke-to-character ratio, or mean number of keystrokes per character ($\overline{kspc}$). This measure is the number of keystrokes, on average, to enter a character under error-free use. It assumes optimal user behaviour, meaning that a shortest path is taken from one character to the next. As $\overline{kspc}$ ignores other factors that affect entry speed, such as visual search, it can only identify *potentially* beneficial designs. A lower $\overline{kspc}$ value does not guarantee superior entry speeds, as the first of two experiments revealed.

Computing $\overline{kspc}$ requires a distance matrix *D* and a digram probability matrix *P*. D contains the distances d_{ij} between the first and second character of every digram *ij*. *P* contains the probability p_{ij} of each digram. The formula is as follows:

$$\overline{kspc} = \sum_i \sum_j (p_{ij} \cdot (d_{ij} + 1)) \qquad (1)$$

where i is the first character of a digram, and j is the second character, p_{ij} is the probability of the digram ij, and d_{ij} is the number of key hops from i to j. The +1 term corresponds to the selection keystroke.

The $\overline{kspc}$ for standard two-handed typing is slightly higher than 1. It is not exactly 1 because of the occasional need to press the shift key *and* another key. It is also clear that the lower bound for $\overline{kspc}$ is 1, since entering a letter in any text entry system requires at least one keystroke.[6] In two-handed typing there are no keystrokes required to travel from one letter to the next, so d_{ij} = 0 for all digrams. Since the sum of all digram probabilities is 1, the formula returns the correct value of 1.

In 0, the $\overline{kspc}$ values of various FOCL design possibilities are compared.

2.4 Learning Model

2.4.1 The Power Law of Practice

Numerous researchers (e.g., Crossman, 1959; de Jong, 1957; Fitts, 1964) have shown that the effect of practice on performance time of perceptual-motor tasks can be approximated by the following power function:

$$T_N = T_1 N^{\alpha} \tag{2}$$

where

T_N	=	performance[7] on first practice session
T_I	=	performance on N[th] practice session
N	=	number of practice time units, and
α	=	the slope of a linear regression line generated from empirical longitudinal data converted to log-log form

---

[6] This excludes word predicton systems such as the *Reactive Keyboard* (Darragh, et al., 1990), which can at times perform text entry in less keystrokes than there are characters, i.e., *kspc* < 1.

[7] The power law has historically been used with performance time data, however, it can be applied to any performance measure, e.g., speed, error rate.

The so-called *power law of practice* also holds for reaction time choice tasks (Seibel, 1963) and a variety of computer-related tasks, such as text-editing (Moran, 1980, cited in Keele, 1986), one-handed typing (Mathias, MacKenzie & Buxton, 1993), pie-menu selection with a stylus (McQueen, MacKenzie & Zhang, 1995) and on-screen keyboard text entry with a stylus (Zhang, 1998). Performance of these and similar tasks, called *speeded tasks* (Keele, 1986), is characterized by a gradually decreasing rate of improvement--the familiar learning curve function of Figure 2-5, below.[8] The power law implies that skill continues to improve *indefinitely*, a claim for which there is some empirical support. For instance, Crossman (1959) found that cigar workers continued to get faster at making cigars after as much as seven years, at which point further improvement was blocked not by human but machine limitations.

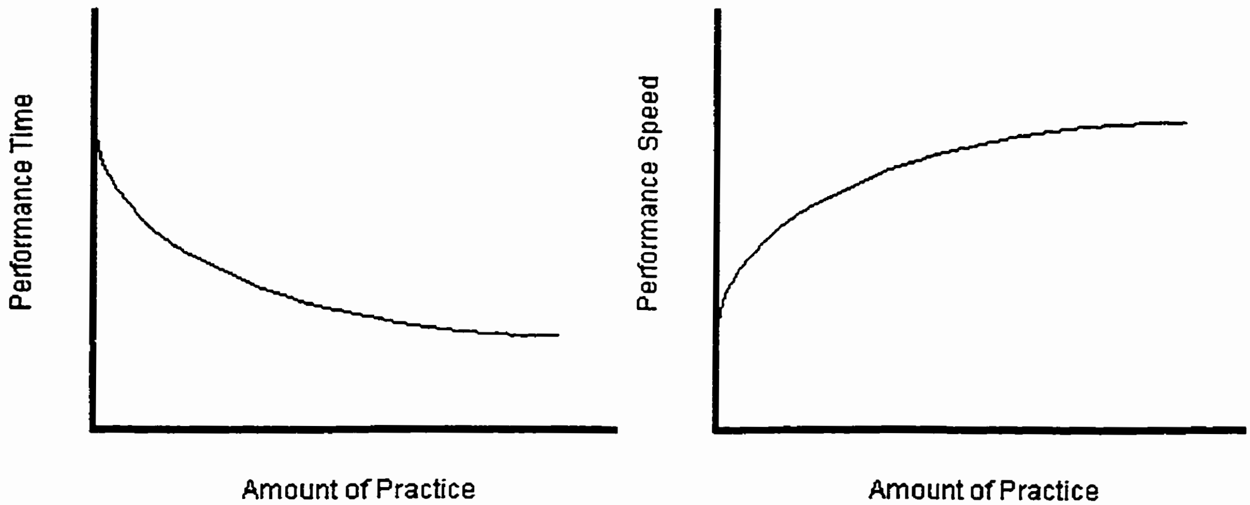


Figure 2-5: Generic learning curve function in its two forms. On the left, time to complete the task decreases with practice. On the right, speed of task completion increases with practice.

Taking the logarithm of both sides of equation (2) produces an equation that is linear in $\log N$. This makes the power law a convenient form for plotting and working with empirical data.

$$\log T_N = \log T_1 - \alpha(\log N) \tag{3}$$

T_I and α are computed by doing a regression analysis of $\log T_N$ on $\log N$ using experimentally collected performance data. The power law is then used to predict performance beyond the amount of practice observed in the experiment. This allows designers of novel systems to evaluate competing designs without performing highly longitudinal studies. It can also help managers determine the cost-effectiveness of adopting a new system, by identifying the amount

[8] Data of this sort are often presented in log coordinates, as the resulting graph is a straight line.

of training required to achieve a desired level of performance. Because of the significant body of empirical evidence supporting it as an accurate approximation of learning (Newell & Rosenbloom, 1981), researchers can be reasonably confident in its predictions. An analysis using the power law of practice is performed on the data from both experiments reported in this thesis.

2.4.2 Novice Skill vs. Expertise

While the power law is in general an excellent tool for modelling learning of speeded tasks, typical performance data tend to deviate systematically from the log-linear function at the two extremes of the regression line. First, subjects' initial performance tends to be better than what the power law predicts. This is because no matter what the task, it is impossible to find a set of subjects with no experience whatsoever at performing--if not the task itself--some transferable component of it.[9] Second, the power law unrealistically predicts that human performance gradually approaches zero. True asymptotes of performance must, of course, be greater than zero.

Newell & Rosenbloom (1981) tried to address these problems with the *generalized power law*, however, this introduced an unrealistically high estimate of initial performance (Keele, 1986).

2.4.3 Skill Acquisition Theories

There are a number of theories that attempt to explain the good fit of the power law to the learning of many skills, and the gradual slowdown in rate that characterizes learning. These theories are summarized in Keele (1986). Crossman (1959) explains performance improvements in terms of the learner's ability to discern more efficient methods with practice and to select those methods faster. Initially, all methods are equiprobable. However, with practice, better methods become more probable and the probability of less efficient methods being selected approaches zero. Thus, the cigar maker with seven year's experience is not faster than his counterpart with 3 years experience because he has had more practice, but because practice has allowed him to see faster ways of performing the subtasks of cigar-making. Crossman's theory adequately explains the slowdown in learning: as better methods dominate, there is less and less room for their probability of selection to increase.

Newell and Rosenbloom's *Chunking theory of learning* (1981) explains learning as the gradual acquisition of a set of stimulus patterns and their associated responses, or chunks. These chunks are hierarchical, that is, smaller chunks (*primitives* in the theory's terminology) combine to form larger chunks. The theory makes several assumptions in order to explain the gradual slowdown in learning. First, the rate of chunk acquisition is assumed to be linear, so that larger chunks take longer to acquire. Second, the frequency of use of a chunk is proportional to its size, so that larger chunks are rarer. With these assumptions, the slowdown is explained as follows: the more skilled the subject becomes the harder it is to get better, because improvement means acquiring larger chunks, which take longer to acquire because (a) they are large, and (b) they are increasingly rare and thus seldom practiced.

An analysis of touch typing according to the chunking theory of learning explains the acquisition of typing skill as follows: (1) The typist initially learns the movements for individual letters; (2) next, the set of movements associated with common short letter sequences are learned as units; (3) over time and with practice, shorter units combine to form longer letter sequences, in particular words. One might argue that the progression from smaller to larger chunks is more discrete, i.e., that the second level of chunks is the word, regardless of size, and that chunks larger than words would not occur.[10] This would have to be tested empirically. The theory also says nothing about the relative difficulty of chunks of the same size. In typing, it is likely that a word requiring an awkward sequence of finger movements to type would take longer to chunk than an equally common word of the same length with a simpler movement sequence. Nonetheless, the theory offers a reasonable account of how typists improve.

Let us submit FOCL-based text entry to a similar analysis. There is a factor of $O(n)$ more primitives than in touch typing, as each letter might appear at n different layout positions. According to the theory, the most commonly occurring will be the most quickly acquired, and this makes intuitive sense. It is likely that larger chunks are acquired even before all the primitives are learned, due to the rarity of many letters occurring in certain contexts. The theory does not contradict this as there is no condition of mastery of *all* primitives before acquisition of larger chunks. We expect common letter sequences with the simplest movement patterns to be chunked soonest. Words such as THE, AND, and THERE, in which each consecutive letter is the most likely to follow its predecessor and is thus placed at the cursor position, may be entered without the use of any arrow keys. Common words with more complex keystroke patterns would take longer to acquire, according to the theory. In this case, unlike in the analysis of typing, there is a better fit with the theory, since longer keystroke sequences actually take longer to enter,

---

[9] MacKenzie & Zhang (submitted for publication) have attempted to capture a more accurate estimate of novice skill level with stylus tapping by randomizing the character layout after each entry.

whereas awkward finger positions are more difficult, but take the same amount of time as simpler ones.

Regardless of the task, the theory predicts that common letter sequences are chunked and eventually incorporated into larger chunks. The problem with this in the case of either typing or FOCL is that it implies that users will identify small chunks within words, irrespective of their position in a word. For example, if the chunk HER were learned then it would be applied not only in the word itself but when encountering words such as WEATHER or CHERRY. Intuitively, this does not seem reasonable because it implies an awareness of a word's constituent *N*-grams that would have to be accessible during reading. For example, CHERRY would be mentally represented as C-H-E-R-R-Y, CH-HE-ER-RR-RY, CHE-HER-ERR-RRY, etc. This is clearly not feasible, in addition to contradicting accepted theories of perception. Gibson's theory argues that we directly perceive words without awareness of their constituent parts (Gibson, 1966), while other theories argue that we learn to read in a bottom-up fashion, but that higher-level knowledge *replaces* lower-level knowledge. When reading is part of the task, it is difficult to accept the theory's prediction of a *gradual* increase in size of chunks. This is made clear by considering that if the strings of letters being entered were meaningless, the gradual increase in size of chunks would be plausible.

In Chapter 4, time data collected in the first experiment is analyzed to determine the effect, if any, of chunking on FOCL entry speeds.

Neither Crossman's theory nor Newell and Rosenbloom's chunking theory adequately account for the increase in response rate to primitive patterns (Keele, 1986). These theories predict an invariant response rate, so that while subjects learn to do more complex things, their reaction time should be the same. In fact, reaction time tends to decrease with practice as well (e.g., Seibel, 1963). Anderson's ACT theory (1982) provides an adequate account of this phenomenon by introducing the notion of the *strength* of a response, which increases each time the response is used. The theory assumes that reaction time--the time to begin executing a response, or what Anderson calls a *production rule*--is proportional to the strength of the response.

[10] Although, it is certainly conceivable that typists chunk common phrases such as "Sincerely yours", or "See you later".

2.4.4 The Crossover Point

The most welcomed result of an evaluation that compares a new method of interaction to an established one is when the new method, though initially inferior due to subjects' lack of experience, outperforms the established method after a moderate amount of practice. The point at which this occurs is called *the crossover point*, and is visualized in Figure 2-6. McQueen, Mackenzie, and Zhang (1995) observed this in their comparison of handwriting and pie menu selection for numeric entry on pen-based computers. Of course, it is conceivable, though rarer, that the new method will perform better than the old one from the beginning, in which case, there is no crossover.

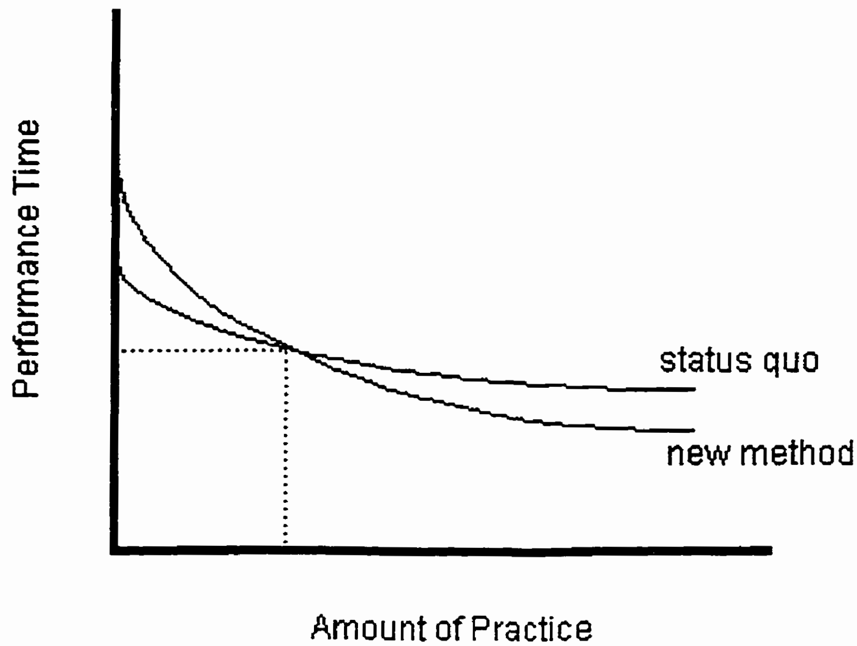


Figure 2-6: The crossover point

Chapter 3: The Design Space

In this chapter we discuss the factors that influence the design of FOCL-based text entry, the ways in which these factors interact, and the tradeoffs that must be considered as a result of their interactions. The resulting design space is surprisingly rich for what, on the face of it, appears to be a simple and straightforward problem. In fact, there are many design paths that might be pursued, and numerous justifiable designs within each path.

3.1 Existing Interaction Styles

We begin this discussion by looking at existing, relevant text entry techniques. In the rush to meet the demand for new features--in this case, text messaging--manufacturers of portable communications devices have failed to take into account the unique requirements of the mobile domain.[11] There has yet to emerge a wide variety of interaction techniques designed specifically for mobile text entry, however, there are domains where designers face similar input and output bandwidth limitations. Three interaction techniques will be analyzed: the telephone keypad method, the date stamp method and the on-screen keyboard method.

3.1.1 The Telephone Keypad Method

The telephone keypad method is well known but little used as a text entry method. There are numerous possible implementations of the method. Two or three, and sometimes four letters of the alphabet are assigned to a number key. With 10 number keys and only 26 letters to accommodate, some keys remain available for additional functions--in standard telephones no letters are assigned to keys 0, 1, * and #. The assignment of letters to number keys, or the *alphanumeric layout*, is arbitrary, but typically the letters are assigned in alphabetical order to successive number keys.

The term *keying logic* refers to how a letter is specified by the user. There are two bits of information to convey to the system, the key on which the desired letter appears and its position

[11] McQueen, MacKenzie, and Zhang (1995) lament the same phenomenon with respect to pen-based interaction.

amongst the letters, or *index*, on that key.[12] In one scheme, the user presses the key containing the desired letter *n* times, where *n* is the letter's index. A drawback of this approach is that it leads to ambiguity in cases where consecutive letters appear on the same key, a problem which must be solved in some way by the interface designer. A common solution is to have a time threshold, after which the system assumes that a new letter is being keyed in. This, of course, slows down entry speeds. Another scheme requires the user to first press the key containing the desired letter, then the number key corresponding to the letter's index. In this approach the system alternates between receiving the key and the index.[13] In yet another scheme the index specification is accomplished by pressing what is effectively an arrow key that moves an invisible cursor. So, for example, if the user has specified the key containing ABC and does nothing further, the cursor is considered to be in the left-most position and the system assumes she has chosen A. If, on the other hand, the user presses the arrow key once, the cursor is moved one position to the right, indicating she wants to select B. If the user presses the arrow key twice she is indicating C as her choice.

Shifting between numeric and alphabetic entry requires a mode shift. This can be provided by a dedicated key that toggles between the two kinds of entry, or separate keys for each mode. The telephone keypad method does not, however, present an obvious way of handling non-alphanumeric characters. Although one can imagine tiny programmable LCDs on each key, or removable labels for indicating key assignments, for all practical purposes the number keys are permanently labeled with their associated letters. Placing more than three characters on a key does not appear to be a path worth pursuing, especially when device size is a strong consideration. For these reasons, support for full-fledged text entry may be an unrealistic stretch of the technique's capability. Note that merely to support purely alphanumeric (excluding punctuation and other symbols) text entry, the space character must be assigned to a key, most likely a dedicated one. This leads to the problem of how to label that key, since there is no obvious symbol for doing so, and key size may not permit the word 'space' to be printed.

While no data on usage of the telephone keypad method for text entry are available[14], it is safe to say that the technique is used, at best, reluctantly by most people. It is difficult to say if this is due to factors other than the interaction technique itself, such as the lack of a display on most phones. Intuitively, the technique ought to benefit from its low keystroke-to-character ratio of

---

[12] This discussion will not cover disambiguation, in which the system guesses the user's intended spelling by comparing all possible permutations to its database. Such an approach works well in situations where the vocabulary is small, but is not viable in free text entry.

[13] Note that there is no reason the index could not be specified before the key.

[14] In one study (Butterbaugh, 1982) subjects did not enter meaningful text.

roughly two-to-one. However, this is probably more than outweighed by the demanding cognitive aspects of the task, characterized by mode switching (Am I conveying key or index information now?), continuous searching (Which key is the letter on?) and mental calculation (What is the index of the letter?). While it might be possible to develop expertise in the telephone keypad method, the technique is not likely to satisfy the requirement of intuitiveness.

3.1.2 The Date Stamp Method

The method is thus named because, as with a date stamp, a desired character must be made 'visible' by rotating a wheel containing the entire character set. As a text entry method, cursor keys allow navigation of the output space, and increment and decrement keys cycle sequentially through the character set at the cursor position. Players of video arcade games are familiar with this technique, which is used to input one's initials into a list of high scorers. Unlike a physical date stamp, selection of a character is performed implicitly by simply moving the cursor to another location. In effect, the cursor position is a moveable editing window of one character. This technique is the standard for entering text in many electronic musical instruments, for example when naming a synthesizer patch.

Two advantages of the date stamp method come to mind. First, there is no need to accommodate the character set in the display, thus freeing up the entire display for output. With a display divided between input and output, the small size of the device necessitates that physical buttons function differently in the two spaces. Such function overlap increases the occurrence of mode errors (Monk, 1986) and is thus undesirable. Second, there is no limit to the number of characters that can be made available, so full-fledged text entry is within the capabilities of the technique. While there is no technological obstacle or size limitation to accommodating large character sets with this technique, very large sets will reduce usability, due to the longer average time to access a character.

The main disadvantage of the method is the very high keystroke-to-character ratio. It is difficult to imagine ways of significantly alleviating this bottleneck. Dividing the character set into logical subsets (e.g., letters; digits; punctuation) would not reduce keystrokes by much, since most time would be spent in the 27 character alphabet subset. Further subdividing the alphabet would only add additional cognitive demands to the task by requiring the user to learn an arbitrary division of the alphabet. The ability to cycle through the character set at various speeds using a type-o-matic[15] feature, or to accelerate through it, may be worth exploring, but is likely to suffer from frequent overshooting errors. Optimizing the ordering of letters would help somewhat, but at the expense of increased learning time. Continually optimizing, or fluctuating, the ordering of letters, as with FOCL, would reduce keystrokes further, but would probably make the experience very frustrating due to the high degree of unpredictability and the difficulty in learning the orderings, since only one character is visible at a time.

3.1.3 The On-Screen Keyboard Method

In this method the display is functionally split into an input and output section. The input portion of the display shows either some or all of the character set. If only part of the set is visible, scrolling, as with the *AccessLink*™ two-way pager (see Figure 1-4), or paging between various subsets, must be supported. Arrow keys move a cursor around the character set, and characters are entered by an explicit press of a select key. The output appears in its own portion of the display.

Because cursor motion is strictly horizontal or vertical, the shortest distance between any two characters may be expressed in terms of a number of keystrokes. Conceptually, the technique is similar to one-finger typing or stylus tapping on a full keyboard, however, the motion restriction is a severe hindrance to faster entry speeds. While stylus tapping on an on-screen keyboard achieves average entry speeds of 30 words per minute (MacKenzie and Zhang, 1998), and speeds as high as 56 wpm have been reported (Zhang, 1998), arrow keys moving a cursor achieve rates of around 10 wpm (Card, English, and Burr, 1978).

3.2 Research Focus

This research aims to increase mobile text entry rates by devising ways to reduce keystrokes. The problem of excessive keystrokes is particularly evident in the date stamp and on-screen methods. However, all three techniques discussed in the previous section could benefit from a reduction in keystrokes. The technique that seems the most promising for improvement by this line of attack is the on-screen method. The keystroke-to-character ratio of the telephone keypad is already quite low, while all the possibilities for reducing keystrokes with the date stamp method carry additional cognitive costs.

---

[15] A feature of software incorporating text entry, in which, if a key is held down for longer than some time theshhold, the corresponding character is repeatedly entered at a constant rate until the key is released.

3.3 Positioning Characters Within the Layout

The basic algorithm for positioning characters in a fluctuating layout is straightforward.

3.3.1 Algorithm

```
Let c = the last character entered
Let A = the alphabetic sequence sorted by order of likelihood to follow to c
Let n = length( A )
repeat i = n times
    place A(i) as close to the cursor as possible
end repeat
```

3.3.2 Discussion

The algorithm does not indicate what to do when there are multiple available positions at equal distances from the cursor. This is, in fact, a design choice. To explore the issue, let us consider a three-row, nine-column layout where the cursor is in the top left corner at the time the layout is rearranged. Figure 3-1 shows a variety of possible ways to position characters according to the algorithm, using the alphabetic sequence corresponding to the space character. The layouts on the right are the result of positioning characters in the sequence atop the figure, according to the patterns on the left.

Are any of these superior to the others? We might argue that, in this specific case, positions in the top row are superior to other positions because they only require the use of one arrow key to reach, whereas positions in rows 2 and 3 require the use of two arrow keys. For example, in the last layout in Figure 3-1, moving the cursor to the letter H in row 1 only requires the right arrow key, while moving to the M in row 2 requires the right and down arrow keys.[16] In other words, positions in row 1 and column 1 are accessible via a less complex route than positions in rows 2 and 3, starting from column 2. It might make sense to fill row 1, and column 1, positions before others at identical distances form the cursor. As a modification to the algorithm, this could be generalized as follows: *Given more than one available position at identical distances from the cursor, positions in the same row or column as the cursor should be filled before other positions.*

---

[16] A counter-argument states that key-repeat rates with the same finger are slower than when alternating fingers are used (Boff & Lincoln, 1988), which would make positions *not* on the same row or column as the cursor more desirable.

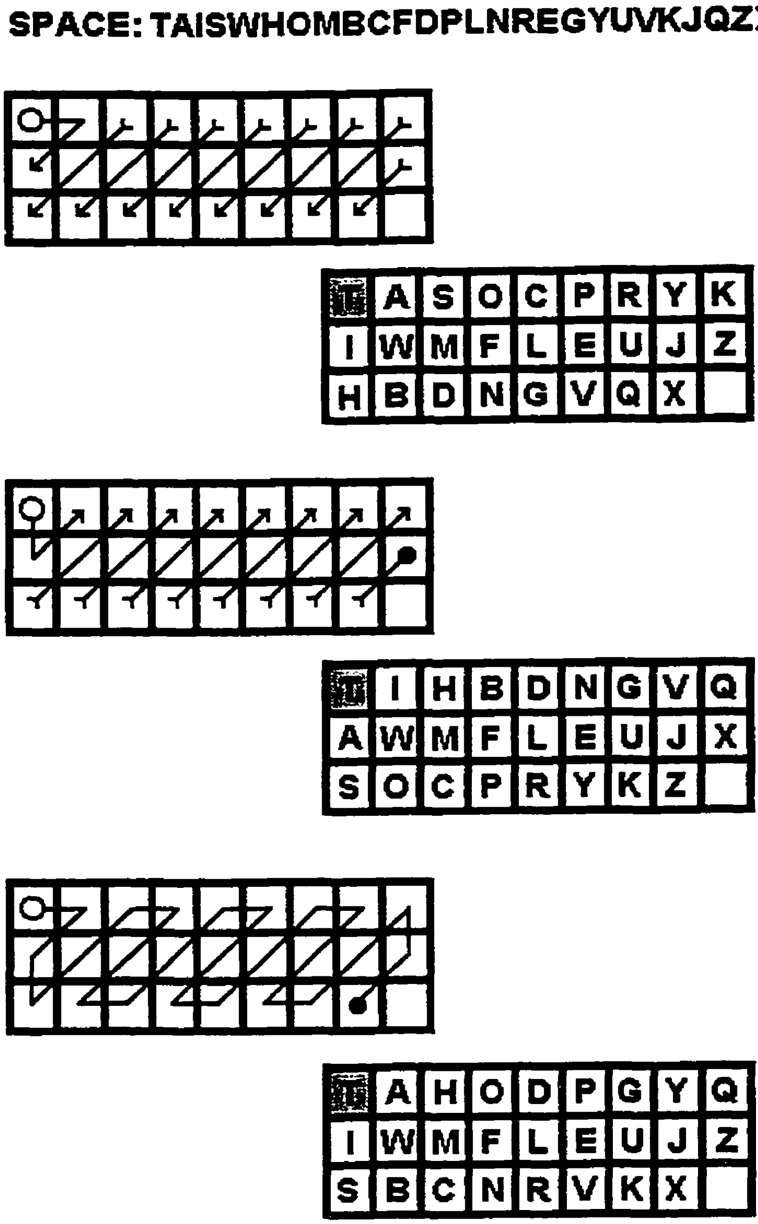


Figure 3-1: Three possible positioning patterns for a 3-row by 9-column layout, and the corresponding layouts following the entry of a space character

Another distinction between identically distanced positions is the number of possible paths from the cursor to the position. For example, in Figure 3-2, both B and C are a distance of three from the cursor, however, there are three possible paths from A to C and only one path from A to B. One could argue that B is better positioned than C because the user has fewer paths to choose from. It has been shown that the number of alternatives affects reaction time (Hick, 1952; Hyman, 1953). As a modification to the algorithm, this could be generalized as follows: *Given more than one available position at identical distances from the cursor, fill positions in increasing order of the number of paths from the cursor to the position.*

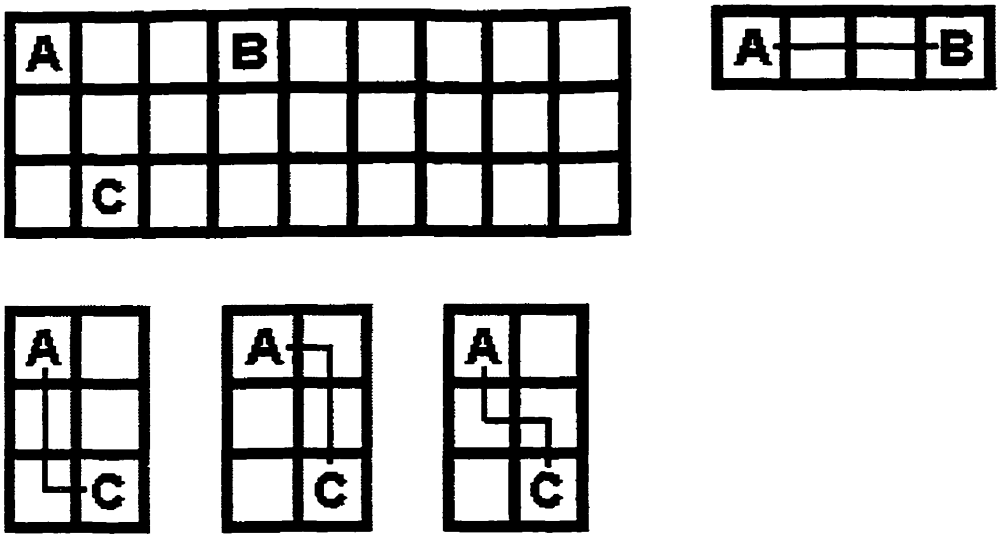


Figure 3-2: Although B and C are both 3 moves away from A, there are 3 possible routes from A to C, but only one route from A to B

Another possibility is that the choice between several identically distanced positions has no significant effect on text entry speed. Determining empirically if this is the case is beyond the scope of this thesis. Experimental results from the literature on visual search offer some direction, however, the application to the particular task of FOCL-based text entry is fraught with caveats on ecological validity and the use of findings from one domain in another.

3.3.3 Visual Search

The positioning algorithm is designed for one purpose: to reduce keystrokes. An unfortunate result of its underlying principle of *proximity to cursor according to probability* is that the layout cannot be easily scanned in an optimal way. To illustrate, Figure 3-3 shows a three-row layout with numbered slots indicating the order of positioning. To search the layout in a way that takes advantage of the way characters are positioned would require a most uncomfortable circular eye motion. The reader is invited to try focusing on the numbers in increasing order.

25	20	14	8	3	9	15	21	26
19	13	7	2	1	4	10	16	22
24	18	12	6	5	11	17	23	27

Figure 3-3: Three-row layout illustrating one possible layout sequence that satisfies the principle of *proximity to cursor according to probability*. The layout requires an unnatural circular eye movement to search optimally.

Alphanumeric arrays are typically scanned from left to right, then top to bottom (Green, Hammond, & Supramaniam, 1983).[17] To attempt to accommodate this by positioning characters in a similar fashion would nullify the main benefit of the FOCL strategy: keystroke reduction. The inherent incompatibility between the way FOCL reduces keystrokes and the way humans perform visual search is *the* critical tradeoff in the design of a FOCL-based system. For the technique to succeed, it is essential to find an appropriate balance between these factors. This tradeoff will resurface at various points in the remainder of the thesis.

3.4 Towards Higher Text Entry Speeds

Merely applying the FOCL technique to a particular layout reduces keystrokes significantly. For example, **Figure 3-4** compares $\overline{kspc}$ (see Section 2.3.2) for single-line fixed and fluctuating layouts. The top part of **Figure 3-4** shows a fixed alphabetic layout, and the top part the particular layout displayed by a FOCL strategy after a space character has been selected. At 4.03, $\overline{kspc}$ for the FOCL layout is 35% less than that of the fixed alphabetic layout at $\overline{kspc} = 6.18$.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

$\overline{kspc} = 6.18$

T	A	W	S	H	F	B	M	C	O	L	P	D	R	N	G	Y	E	I	K	U	V	J	Q	Z	X

$\overline{kspc} = 4.03$

Figure 3-4: Comparison of kspc values for single line character set displays using FOCL (fluctuating) and alphabetic (fixed) layouts

If no other factors affected entry speed, then this in and of itself would lead to higher entry rates. Although comparing $\overline{kspc}$ values of possible designs is useful in reducing the set of viable alternatives, a lower $\overline{kspc}$ value does not necessarily imply a better (i.e., faster) design. Other factors affect entry rates. In particular, rearranging the layout after each character entered adds visual search time to the task, as the user must locate the next desired letter in an unfamiliar layout.

FOCL will only outperform a fixed layout approach if the savings gained by reduced keystrokes outweigh the cost of added visual search time. Put differently, it is the job of good design to manage this tradeoff so that the negative effect of visual search time on entry speed is does not

[17] There is, no doubt, a Western bias in this finding. It is unlikely that readers of languages not written from left to right would exhibit the same behaviour.

cancel out the positive effect of reduced keystrokes. Two important design goals for FOCL-based text entry emerge from this statement.

- **Goal 1:**

  Enhance the potential of users to acquire familiarity with the layouts

- **Goal 2:**

  Reduce keystrokes as much as possible

The goals themselves compete with one another, a fact the designer must be sensitive to. For example, while fluctuating the layout reduces keystrokes (satisfying Goal 2), it increases the number of layouts, creating a system that is more difficult to learn (violating Goal 1). The next two sections address the two design goals.

3.4.1 Affording Familiarity with Layouts

Reducing The Number Of Possible Layouts

The advantage of a fixed layout over a fluctuating approach is that users can easily become familiar with the layout. Visual search time effectively drops to zero. On the other hand, a fluctuating layout is much more difficult to learn.

For a character set of size n, there are $O(n^2)$ possible layouts since there are n alphabetic sequences, n positions in the layout, and each sequence may be arranged optimally around any of the positions. With the limited set of 27 characters, there are 729 possible layouts. Although some layouts never occur, and some are far more common than others, the set of layouts is difficult to learn. Note that if trigrams (third order optimality) were the basis for generating layouts, this number would increase to $27^3 = 19683$. If tetragrams (fourth order optimality) were used there would be $27^4 = 531441$ possible layouts.

This assumes, however, that characters are arranged about the position of the last character selected. By snapping the cursor to a fixed home position after each character entry, the number of possible layouts is reduced from $O(n^2)$ to $O(n)$, a feature that significantly improves ease of learning. The 27 layouts of the first experiment prototype are shown in Appendix A, on p. 110.

Chunking

If we accept the chunking theory (p. 28) as applicable to FOCL-based text entry, then a number of design principles with the potential to facilitate chunking emerge:

- **D1:**

  Reduce the number of different positions in which a letter occurs

- **D2:**

  Find ways of reducing the time to enter letters distant from the cursor

One of the drawbacks of FOCL, noted by many subjects in both experiments conducted[18], is the uncertainty of an upcoming letter's location in the layout, which leads to hesitation and errors. D1 is a way of reducing this uncertainty. With increasing confidence in predicting the position of letters, users hesitate less and make fewer errors, making practice of common keystroke patterns more efficient. This speeds the learning of chunks, according to the chunking theory. Paradoxically, D1 is an effort to make FOCL more like a fixed layout, and so can only be applied to the point before which the benefits of a fluctuating layout are lost.

Let us define *close to the cursor* as being no more than two keystrokes away. A letter must to be no greater than the eleventh letter in the alphabetic sequence used to generate a layout to satisfy this condition for a three-row display, as Figure 3-5 illustrates, because in the best case, the cursor is totally enclosed, leaving room for 11 positions within two keys of the cursor. There are many common words in which most letters appear close to the cursor, but one or two do not. In the word APPLE, for example, using the alphabetic sequences in Table 2-4 (p. 23), P is only the 15th most likely letter to follow A. A, L and E are the second most likely to follow space, P, and L respectively, while P is the ninth most likely to follow P. If the time to enter "outliers" such as the first P in APPLE could be reduced, for example, by providing a way to quickly jump to the extremities of the layout, the chances of such words being chunked would increase, according to the theory, because the chunks would be smaller. This is one way of applying D2. Another way of applying it is the use of higher-order *N*-gram data, which would tend to reduce the occurrence of outliers altogether.

---

[18] Discussed in Chapter 4 and Chapter 6.

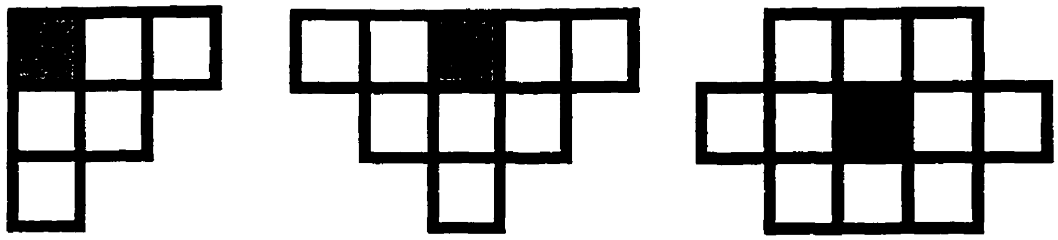


Figure 3-5: Various cursor scenarios and associated number of layout positions at a distance of two or less keys from the cursor.

It is hoped that over time users can develop a larger and more complex repertoire of chunked patterns in FOCL text entry. However, this was not widely observed in the two experiments, and is discussed in the next chapter.

3.4.2 Design Factors Affecting *kspc*

We have seen, in the comparison of one-row fixed and FOCL keyboards (Figure 3-4), that merely fluctuating a layout by applying digram probabilities reduces keystrokes. $\overline{kspc}$ can be reduced further by manipulating any or all of the following design variables:

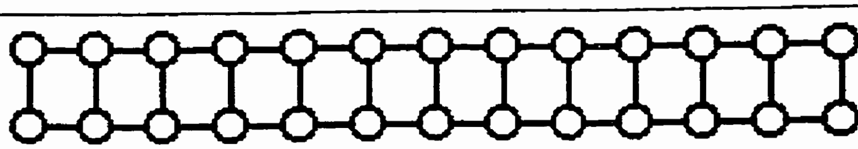
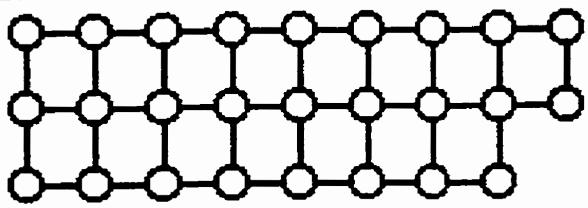
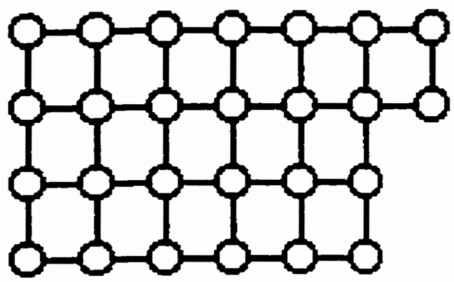
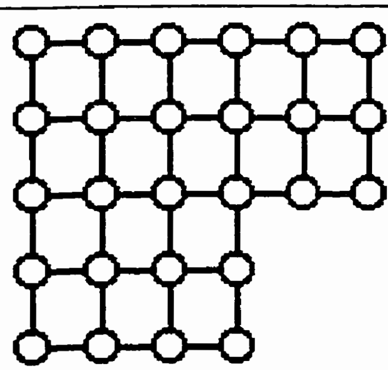
- The number of rows in the layout
- The location of the cursor home position
- The treatment of the space character

Each of these is discussed in turn.

Number of Rows

Increasing the number of rows in a keyboard layout increases the number of adjacent keys. To formally illustrate this, we represent the shape of the keyboard as a graph, that is, a set of nodes and a possibly empty set of edges connecting them (Gould, 1988). To avoid confusion with the notion of the *edge of a layout*, by which we mean an outer side or extremity, the term *arc* will be used in place of edge. In the graph representation of a layout, the character positions are its nodes and an arc exists between any two adjacent nodes. Thus, we can define a cursor move as an arc traversal. Table 3-1 shows graph representations for two, three, four and five row layouts.

Table 3-1: Graph representations and average inter-node distances for two, three, four, and five row 26-character on-screen keyboard layouts.

Rows	Graph Representation	Average Inter-Key Distance
2		5.0
3		4.0
4		3.67
5		3.67

The more arcs in a graph, the more *connected* it is. The more connected it is, the fewer arc traversals required, on average, to get from one node to another. The right column in

Table 3-1 shows average inter-node distances for the various layouts, which decrease with increasing number of rows. Since the character set is finite, however, there is a point at which increasing the number of rows begins to decrease the number of arcs. This is the point at which there are more columns than rows, and we have essentially rotated a previously visited layout by ninety degrees.

The maximum connectedness of any node is four, corresponding to the number of arrow keys. It is conceivable that a design could allow movement in more than four directions. In this case the characters would have to be enclosed in *N*-sided shapes, where *n* is the number of directions. This renders the notion of rows somewhat sterile, however, as keys would no longer line up neatly in discrete rows, but rather partially overlap. Figure 3-6 shows a sketch from the patent application for Mobile Client Computer with Hexagonal Keyboard, which uses hexagonal-shaped keys (Bertram, Champion, & Eichorn, 1998). It might be difficult, however, to implement this approach on the low-resolution, line- and character-based LCDs of mobile devices.

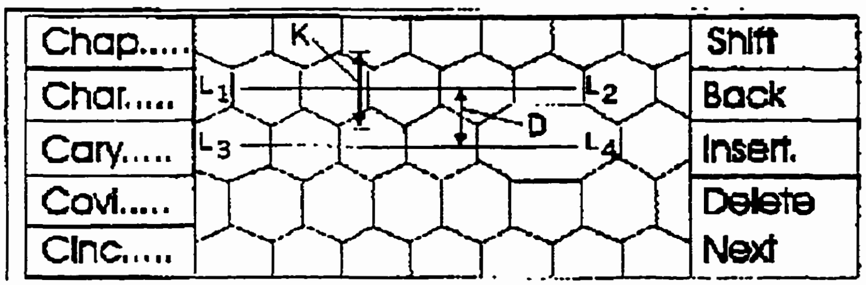


Figure 3-6: On-screen keyboard with assignable hexagonal keys. (Bertram, Champion, & Eichorn, 1998)

Cursor Home Position

Any of the positions in the layout may be selected as the cursor home position, however, to establish the effect on $\overline{kspc}$ three types that capture the variability of all possibilities are used: At a corner (Figure 3-7a); at an edge, but not at a corner (Figure 3-7b); in the centre (Figure 3-7c).

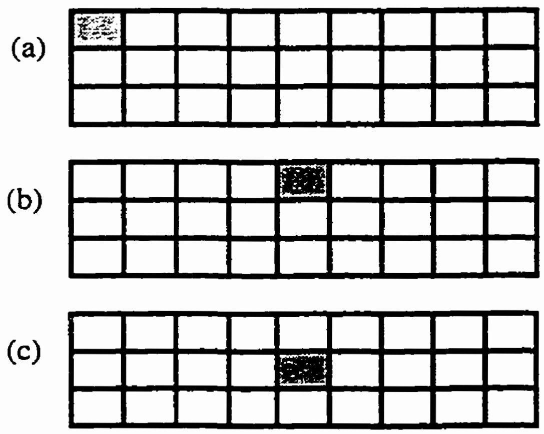


Figure 3-7: Instances of each of the three categories of cursor home position used in analysis: a) corner b) edge, but not at a corner c) centre.

Table 3-2 shows each of the cursor positions in two, three, four, and five row layouts. The number in each square is the distance between that square and the cursor. The average distance from the cursor to a key, computed by averaging the numbers in all the squares, is lowest for a centre cursor, and highest for an edge cursor. This is the same as inter-node distance, except that we are never required to move from one letter to another, but only from the cursor to any key. Again, when we factor digram probabilities in, the effect is stronger. All other things equal, $\overline{kspc}$ is lowest for centre cursor layouts, and highest for corner cursor layouts.

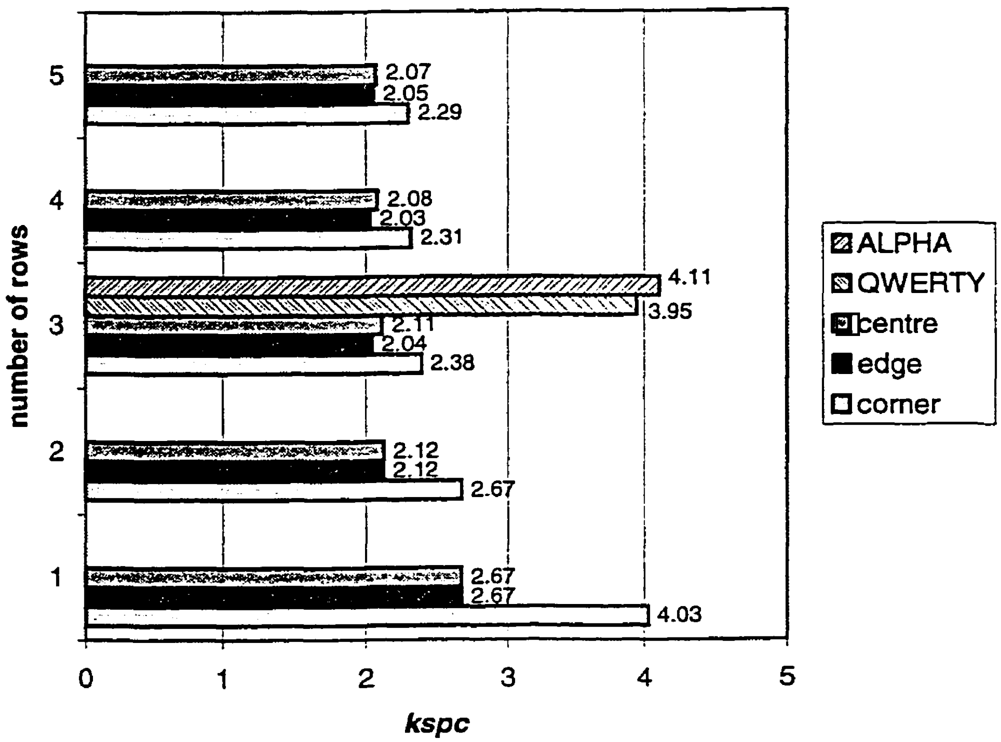
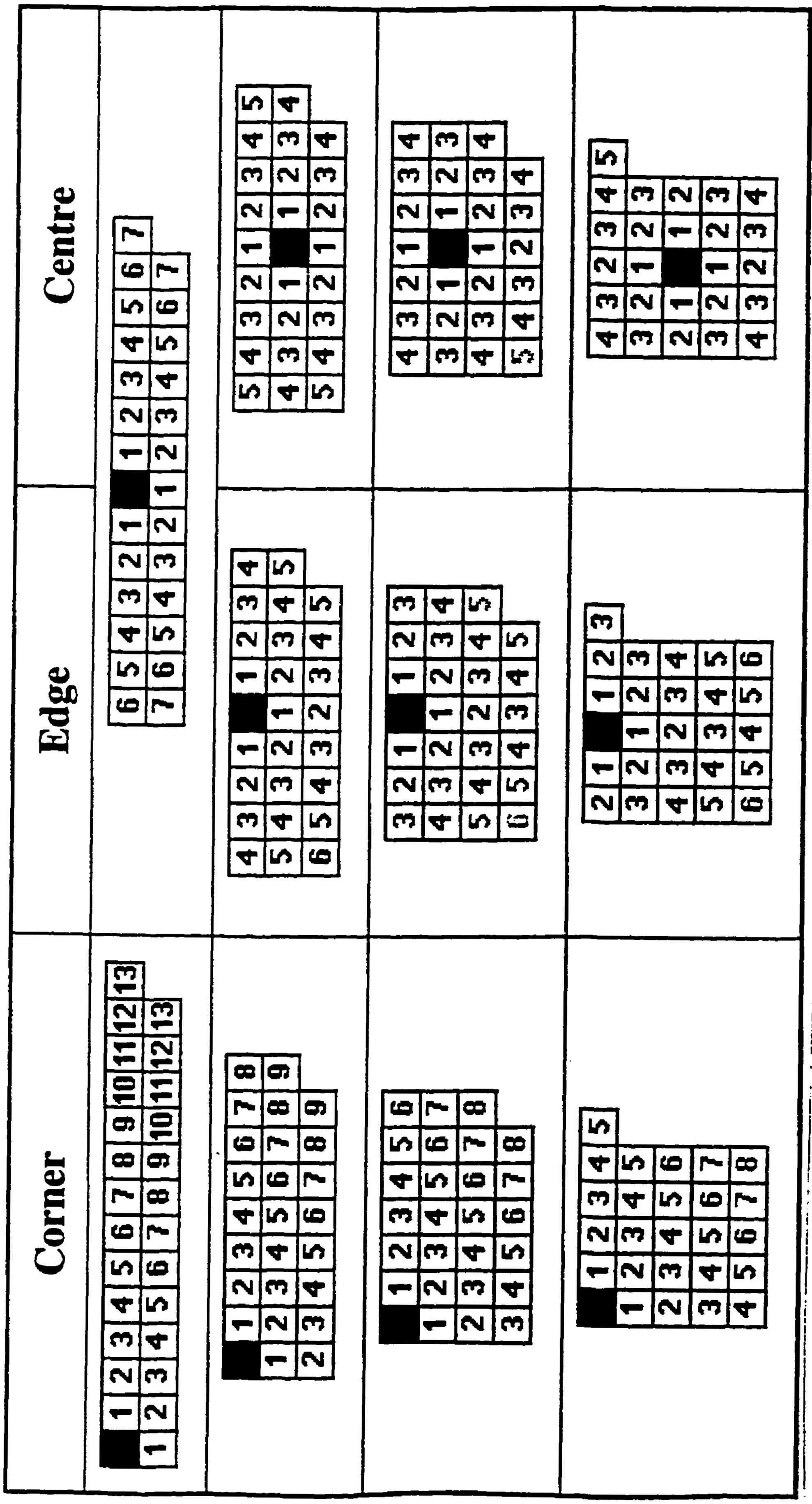


Figure 3-8: Graph of keystrokes per character ($\overline{kspc}$) by number of rows and type of cursor home position.

Figure 3-8 summarizes the combined effect of number of rows and cursor home position on $\overline{kspc}$. $\overline{kspc}$ values for fixed QWERTY and fixed alphabetic three-row layouts are included for comparison. The only difference in interaction style between the fixed and fluctuating layouts used to compute the data is that the cursor in the fixed layouts does not snap to a home position after each character entered.

Table 3-2: Two, three, four, and five row layouts with corner, edge, and centre cursor home positions. Numbers in cells indicate distance to cursor position.



The graph shows that $\overline{kspc}$ tends to decrease as the number of rows increases. There is also a noticeable effect of cursor home position category on $\overline{kspc}$. Regardless of the number of rows, corner cursor designs have higher $\overline{kspc}$ values than centre cursor designs, which in turn have higher $\overline{kspc}$ values than edge cursor designs.[19] In addition, $\overline{kspc}$ for any of the 3-row fluctuating layouts is dramatically lower than the $\overline{kspc}$ values for either the alphabetic or QWERTY fixed layout. In fact, with the exception of the 1-row corner cursor layout, all the fluctuating layouts have lower $\overline{kspc}$ values than the two fixed layouts. It is somewhat surprising that centre cursor designs have higher $\overline{kspc}$ values than edge cursor designs. This turns out to be due to distance of two from the space character, as discussed in the following section.

Treatment of the Space Character

The space character receives special treatment in the physical keyboard due to its overwhelming prominence in typing. In English, the space character occurs with a probability of .18 (Soukoreff & MacKenzie, 1995). The next most common character, E, has a probability of .11. Clearly, the treatment of the space character in a FOCL-based system is critical, both in terms of keystrokes and visual search. A design choice that reduces the number of keystrokes to enter a space by half, will lead to roughly a 10% reduction in overall keystrokes. A design choice that eliminates searching for the space character, by fixing its position, will significantly reduce visual search time.

Should the space character receive special treatment, or simply fluctuate along with the other characters? On the standard keyboard, the space bar's size and position under the thumbs make it easily accessible at any time. It is worthwhile emulating this aspect of the physical keyboard in an on-screen keyboard. Two ways of affording special status to the space character emerged in the design process. The first is to always place the space character at the cursor position. The second is to place the space character outside the layout but adjacent to the cursor home position. With either of these solutions, the space character can be within a single keystroke after a character selection.

There are additional design issues in each of the above solutions. Treating the space character the same as the other characters leads to problems in representing it within the layout. By a blank? By some special character? Since the representation of the space character is nothing, alternative representations might not be easily located in a search task. If the space character is always

[19] In one- and two-row layouts edge and centre cursors are identical.

positioned at the cursor position, on the other hand, the benefits of chunking may be lost. Letters would never appear at the cursor position and would thus always require use of the arrow keys to select. It is far easier to learn and execute a routine of consecutive presses of the same button than it is to execute a sequence of different button presses.

If the space character is placed outside the layout, adjacent to the cursor home position, a number of issues arise. Where will it be located? With a corner cursor there is a choice of the two outer sides of the cursor key (Figure 3-9a). With an edge cursor there is only one available side, so there is only one choice (Figure 3-9b). With a centre cursor, this solution must be modified, since the cursor is not adjacent to any of the edges of the layout (Figure 3-9c). The necessary modification is that two keystrokes are required to select a space. Is this acceptable given the prominence of the space character? One could argue that while it doubles the keystrokes, the same button has to be pressed, and doing so in quick succession may not take twice as long.

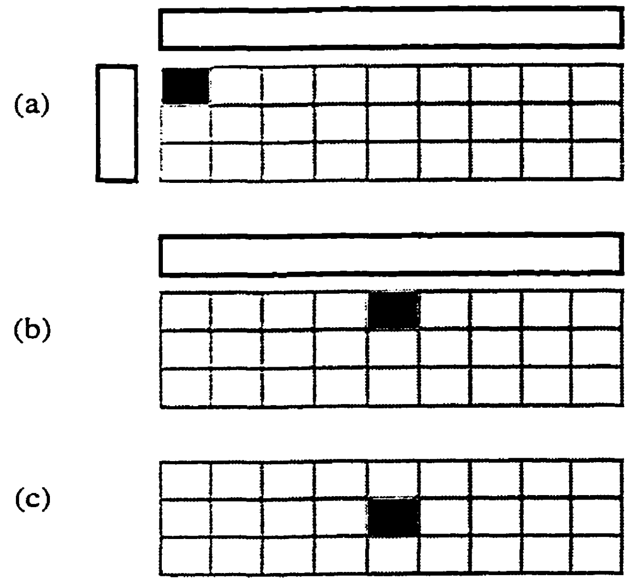


Figure 3-9: Possible placements of space character in layouts using (a) corner, (b) edge, and (c) centre cursor home positions, so that the distance to the space character is only one keystroke. The black rectangles are where the space character could be positioned.

Implicit vs. Explicit Selection

In the description of the date stamp method, the concept of implicit selection was introduced. In that style, no keystroke is required to select a given character. Simply moving the cursor away from its current position implies a selection. Given the high frequency of occurrence of the space character, eliminating the selection keystroke would have a significant effect on overall keystrokes. Using the layout shape and centre cursor home position in Figure 3-9a to illustrate, one way to implement this is with a single left or up arrow stroke (towards the space key) that would momentarily highlight the space, then snap the cursor back to its home position.

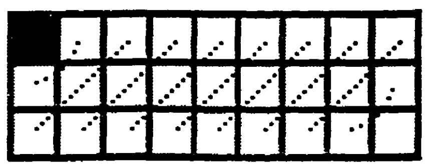
3.4.3 Tradeoffs

As in any design space, the effects of manipulating one design factor are not independent of other factors. In this section we discuss some of interactions between two or more design factors, and the resulting tradeoffs.

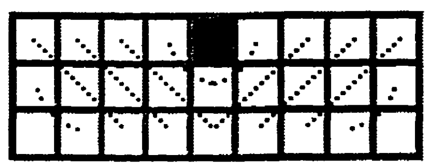
Cursor Home Position and Optimal Search Strategy

Section 3.3.3 introduced the issue of awkward optimal visual scanning patterns dictated by the FOCL character positioning algorithm. Figure 3-10 depicts the optimal scanning patterns associated with each of the three categories of cursor home position. Although none of the patterns can be described as natural, pattern (a) is arguably the least unnatural of the three, since the scan angle is only 90 degrees, as opposed to 180 degrees for (b) and 360 degrees for (c). The pattern in (a) is also closest to a left-to-right, top-to-bottom search pattern.[20] Pattern (c) is clearly the worst in this respect, as it necessitates a concentric circular scan pattern. Thus, the centre and edge cursor home positions are poorer choices than the corner position from the perspective of visual search.

(a)



(b)



(c)

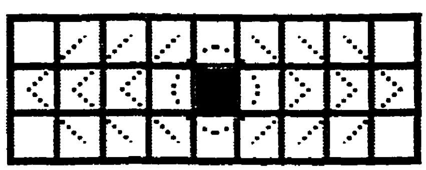


Figure 3-10: Optimal visual scan patterns associated with a) corner, b) edge, and c) centre cursor home positions.

[20] Although this is due to its precise positioning in the top-left corner, and would not be a feature of other instances of the corner cursor home position.

Cursor Home Position and Movement Decision Time

Before moving the cursor towards a letter, a FOCL user has to decide which of the possible directions to move in and which key to press. With a layout incorporating a corner or edge cursor home position, the possible directions are constrained to two and three respectively, thus simplifying the decision (Figure 3-11). Decision time is proportional to the number of alternatives (Hick, 1953; Hyman, 1953), which represents an advantage for the corner cursor.

In addition, assuming ideal usage (i.e., no overshooting of target letters or backtracking) the number of arrow keys to navigate the character set is also two, three and four for corner, edge, and centre cursor home positions, respectively. With less than the full array of arrow keys needed, the choice of either a corner or edge cursor home position frees up the idle keys for additional functions. In particular, one of these idle keys can function as a dedicated space bar. Again, though superior with respect to keystroke reduction, a centre or edge cursor home position is not the best choice under this criterion.

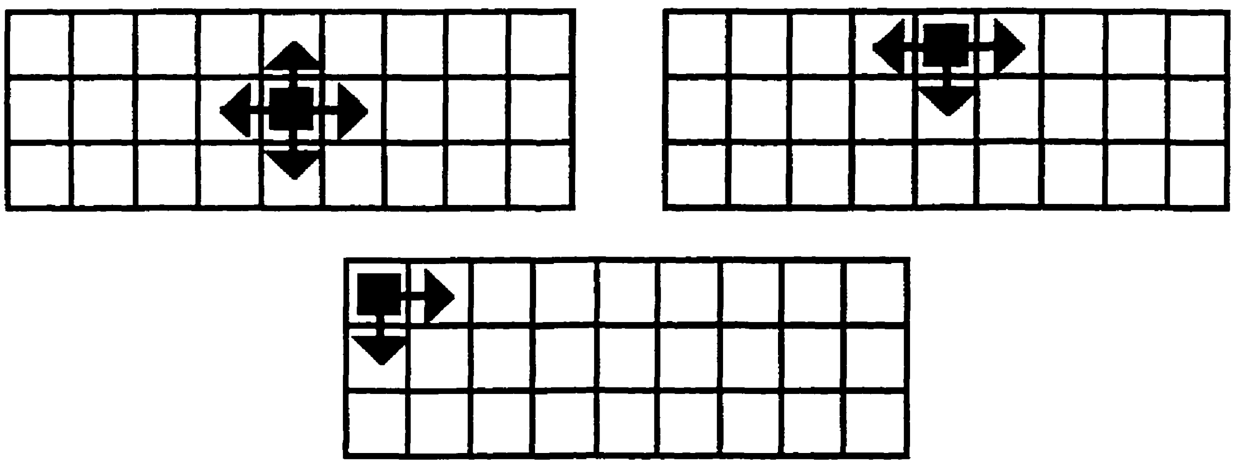


Figure 3-11: The number of possible directions, assuming error-free usage, associated with each of the three cursor home position categories.

Location of Space Character and Cursor Home Position

A centre cursor home position is cut off, as it were, from the sides of the layout. Therefore, a dedicated on-screen space key would be at least two keystrokes--depending on other aspects of the layout--from the cursor, as opposed to one keystroke for layouts with cursor home positions adjacent to the edge of the layout (i.e., corner or edge cursors). Positioning the space within the layout, like all other characters, is not a good solution unless implicit space character selection is abandoned. Without doing so, there would be numerous errors in cases where the path from the cursor to a letter passed through the space position. Another possible treatment is to always put the space character at the cursor home position, however this would forfeit the possible benefits of chunking. According to this logic, a centre cursor home position conflicts with virtually all of the desirable space character treatment features discussed on page 47.

Implicit Space Selection and Cursor Home Position

A potential problem with implicit selection of the space character is that it may confuse the function of the arrow key used to access the space bar, possibly leading to a higher error rate. It also introduces an inconsistency into the interface--now one character is selected in a different way than all the others. This issue further recommends either a corner or edge cursor home position, each of which, in error-free usage, have at least one arrow key that is unused. One of these idle keys may be treated as a dedicated space bar. Under the reasonable assumption that users approach error-free usage over time, the confusion and inconsistency described above is eliminated, with either a corner or edge cursor home position.

3.4.4 The First FOCL Prototype

In order to explore this interaction technique and test its viability experimentally, a prototype was created in Macromedia's *Director*. The software and hardware setup of the prototype are described in the next chapter. Figure 3-12 shows the design choices for the prototype, including a 3-row layout, top-left corner cursor home position and space character to the left of the layout.

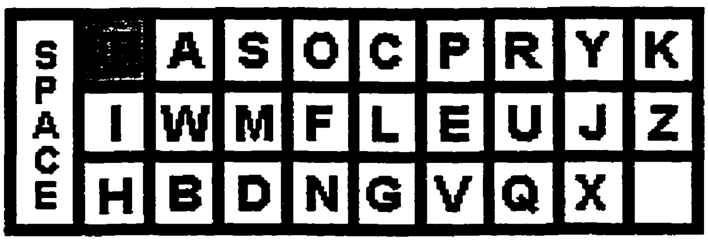


Figure 3-12: First FOCL prototype, showing top-left cursor home position, three-row, nine-column shape, and space character located to the left of the layout. The layout shown appears after entering a space character.

3.4.5 Design Choice Rationale

Number of Rows

The choice of a three-row layout is based on a conservative estimate of four lines as the typical display height of portable communications devices, exemplified by the Ericsson *PTE 218* in Figure 3-13. With one line reserved for output, this leaves three for the character set.

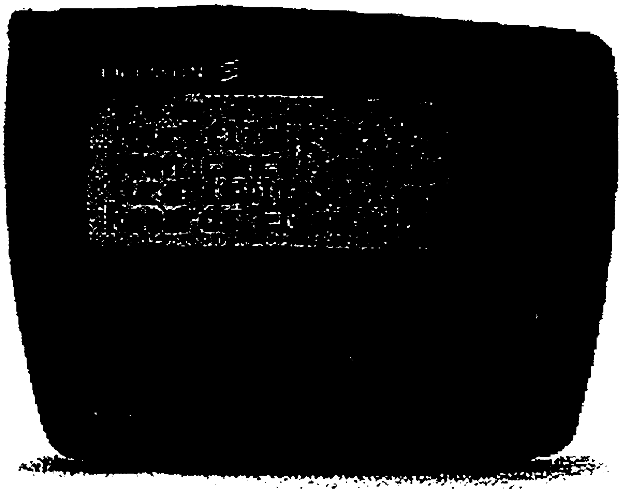


Figure 3-13: The Ericsson *PTE 218* pager. (from http://www.ericsson.com)

Recently, pagers with as many as eight display lines have been marketed, for example, the *Inter@ctive 900* and *950,* by Research In Motion (Waterloo, Ontario, see Figure 3-14), so perhaps this decision should be re-evaluated. The choice of a three row layout was also an attempt to match the QWERTY layout, used as the fixed layout condition in the first experiment. This eliminated a possible source of confounding.

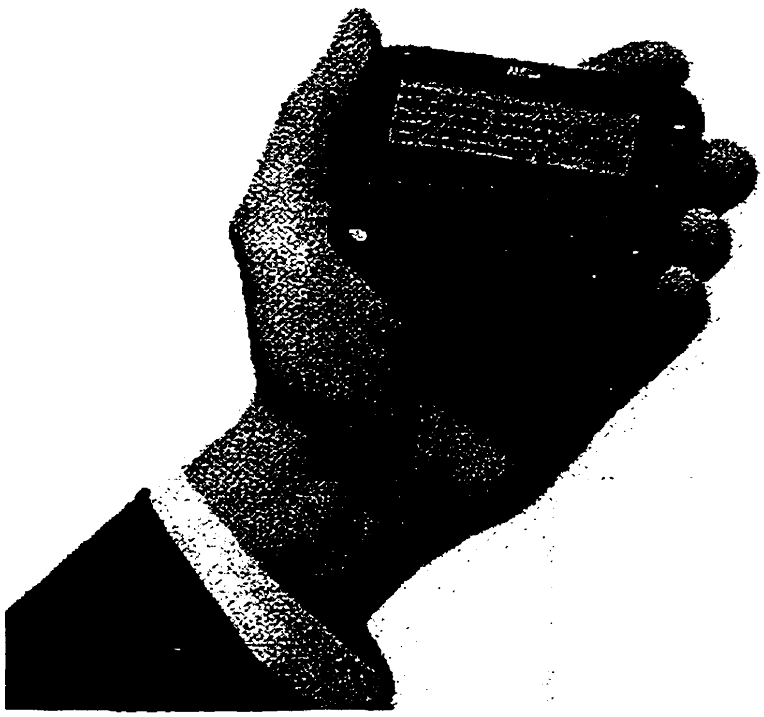


Figure 3-14: The RIM *Inter@ctive 950* two-way pager, with eight-line display. (from http://www.rim.net)

Cursor Home Position

Edge cursor home positions provide the most keystroke reduction, though the difference in $\overline{kspc}$ between edge and centre cursors may be negligible. A centre cursor does not afford a dedicated space key, as it is not adjacent to the edge of the layout--a distinct disadvantage. This requires a mode shift when entering a space, which will lead to more errors (Monk, 1986), Another consideration is the optimal scanning pattern dictated by the cursor home position. From this perspective, the corner cursor, by reducing the angle of scanning to 90 degrees from 360 degrees for a centre cursor and 180 degrees for an edge cursor, is the best choice. Another advantage of a corner cursor over an edge cursor is its affordance of two-key navigation, as opposed to three-key, reducing decision times surrounding the direction of the next move. For these reasons, corner cursor design was chosen.

Having chosen a corner cursor, the choice of top left corner followed naturally in that it is consistent with initial cursor position expectations for English text entry. The choice of placement of the space key to the left of, as opposed to on top, of the layout was arbitrary. There was no way, given the top-left cursor position, to place the space character at the bottom, as in the QWERTY layout.

Chapter 4: First Experiment

4.1 Method

An experiment was conducted to evaluate the FOCL technique. The experiment was exploratory in nature, serving to identify key issues in the design of FOCL-based text entry. To determine what, if any, performance benefit could be derived from optimally fluctuating the layout, the prototype (see Figure 3-12, p. 51) was compared to a fixed QWERTY layout using the same interaction style. The experiment proceeded with the following hypotheses:

Hypothesis 1: Entry speeds in the FOCL condition will exceed those in the fixed QWERTY condition after a moderate amount of practice, i.e., a crossover point (see p. 31) will be observed.

Hypothesis 2: Improvement in FOCL entry speeds over time will be partially due to chunking (see p. 41).

Hypothesis 3: The further a target letter from the cursor in the FOCL condition, the longer it will take to locate. The time to locate a target letter in the fixed QWERTY condition will not be affected by distance from the cursor.

There was no hypothesis regarding error rate.

4.1.1 Subjects

Subjects were solicited by both electronic newsgroup postings and paper notices placed around the University campus (see Appendix B, p. 111). The only requirements for participation were strong English proficiency and availability on consecutive days until completion of the last session.

Twelve people who responded to the solicitations were admitted into the study, however one subject did not appear for the first session, and after several unsuccessful attempts to reschedule, was dropped. Thus, a total of eleven subjects, six male and five female, participated in the experiment. With one exception, subjects were University of Toronto students from a variety of disciplines. Eight of the ten students were undergraduates, the remaining were graduate students. The lone non-student was trained in a professional occupation. The age of subjects ranged from 20 to 35, with an average age of 25.8 years.

Though it was not a requirement for participation, all eleven subjects were right-handed. Eight of the eleven subjects were native English speakers. The three non-native English-speaking subjects were fluent in English. All but two subjects possessed touch-typing skill. This information was collected to check for an effect of familiarity with the QWERTY keyboard on entry speed in the fixed condition. Subjects were paid $80, or $8 per session, upon completion of their last session. All eleven subjects completed the experiment.

4.1.2 Apparatus

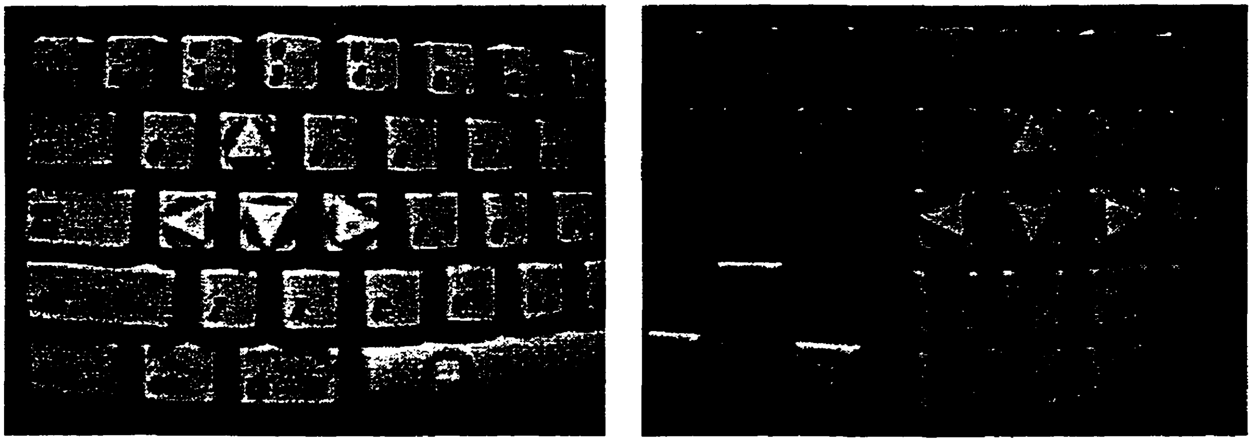


Figure 4-1: Navigation and selection keys for left- and right-handed subjects, on standard Macintosh keyboard used in experiment.

The experiment was conducted using a Power Macintosh computer with a 19" display. Input was achieved using a standard Apple keyboard with labels on certain keys (see Figure 4-1) chosen to function as arrow keys. The criteria for choosing these keys were (a) that they form an inverted-T shape, and (b) that when the index, middle and ring fingers are placed on the horizontal part of the inverted T, there is another key underneath the thumb (see Figure 4-1). The first criterion mimics the common orientation of the arrow keys found on many keyboards, enabling users to navigate the character set without looking at their hand. The second criterion ensures that the selection key is in a comfortable position for the thumb.[21] For right-handed subjects, the 4, 5, 6 and 8 keys on the numeric keypad were labeled as left, down, middle and up arrow keys, respectively, and the left arrow key was designated as the selection key (Figure 4-1, right side).

For left-handed subjects--had there been any--the a, s, d and w keys were labeled as left, down, middle and up arrow keys, respectively, while the space bar was designated as the selection key (Figure 4-1, left side).

Subjects were seated in a chair with their head at a distance of approximately 75 cm from the monitor. A keyboard rest was provided to prevent discomfort to the hand performing text entry.

The software was written in *Lingo*, the programming language within Macromedia's *Director*. *Lingo* is a declarative fourth generation programming language with an English-like syntax. While Director originated as a presentation and prototyping tool, it has grown into a full-fledged programming environment incorporating such features as object-oriented programming. It was chosen as a development environment because of the author's previous experience and strong familiarity with the software.

The labeled arrow keys controlled a cursor on the screen, allowing users to navigate the character set. In addition to the character set, the screen consisted of a large text field at the top of the screen for displaying stimulus phrases, and a smaller field for displaying output, as seen in

Figure 4-2.

To eliminate confounding effects, efforts were made to match the FOCL condition to the QWERTY condition as much as possible. For instance, as mentioned in the previous chapter, a three-row layout was chosen to match that aspect of the QWERTY layout. However, certain features of the QWERTY layout could not be transferred to the FOCL condition, either because previous design choices had ruled them out, or because they would have a negative effect on performance. The software, therefore, differed between the two conditions in the following ways:

- In the fixed condition, the cursor did not snap to a home position after each character entered, as it did in the FOCL condition.

- The space character was below the character set in the fixed condition, while in the FOCL condition it was to the left of the layout. The positioning of the space character in the FOCL condition was constrained by the decision to use the top-left corner as the cursor home position.

---

[21] The arrow keys themselves do not satisfy the second criterion as they are already at the bottom of the keyboard.

- Because the cursor snapped to a home position, the space character was never more than a keystroke away from the cursor in the FOCL condition. In the fixed condition, however, it could be anywhere from one to three keystrokes away from the cursor after a character selection.

- The fixed layout had row lengths of ten, nine, and seven characters, corresponding to the QWERTY layout on a physical keyboard, whereas the FOCL layout had rows of nine, nine, and eight characters, to reduce the number of isolated keys in the layout.

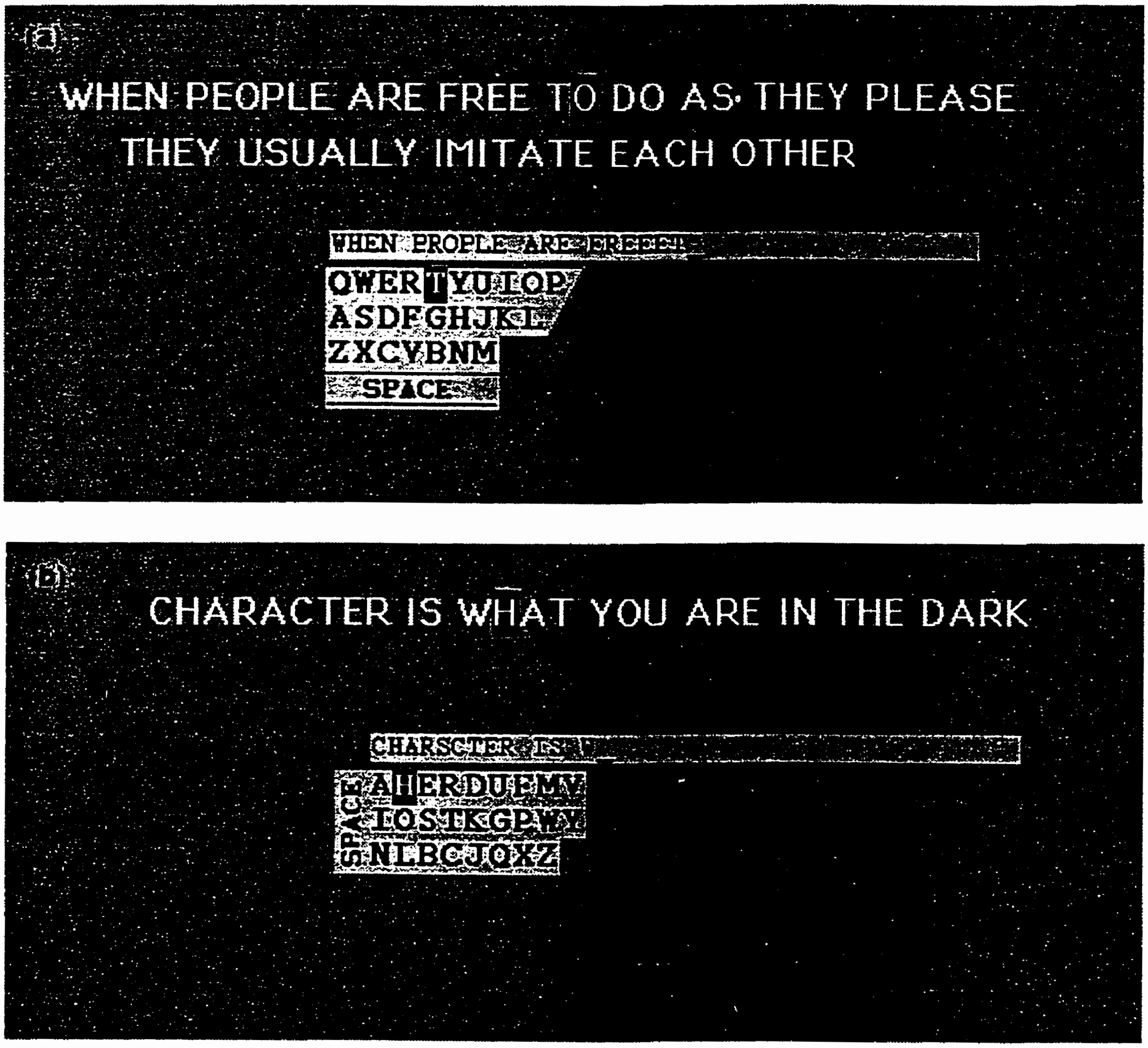


Figure 4-2: Screen shots of first experiment software, during (a) the fixed QWERTY condition and (b) the FOCL condition. The stimulus phrase is at the top, the character set at the bottom, and the output line just above the character set.

4.1.3 Procedure

Before the first session, subjects read an information page (Appendix B, p. 112) describing the nature of the study, the experimenter's expectations of participants and obligations of the experimenter regarding payment and confidentiality. Subjects signed a consent form indicating their understanding of the conditions of participation. Before starting the first session, subjects completed a software tutorial accompanied by written instructions (Appendix B, p. 113), lasting approximately ten minutes. The tutorial familiarized subjects with the two conditions and the various nuances of the interface. It did not allow subjects to practice extensively. At the end of the first session, subjects completed a short demographic questionnaire (Appendix B, p. 115).

To begin a session, subjects would press any key on the keyboard, causing the first phrase to appear. A new phrase would appear after each completed phrase, until the subject had performed 15 minutes of text entry, not including timing pauses during the intervals between the last character entry in a phrase and the first keystroke of the next phrase. The software would not interrupt a subject in mid-phrase at the end of a condition, so that the length of each condition was slightly more than 15 minutes.

Subjects were instructed to read a phrase in its entirety and then begin entering it. The phrase remained on screen until completion, so that subjects could consult it again if necessary. To avoid long strings of consecutive errors due to subjects losing their place within a phrase, the software highlighted the next character to enter in the phrase, as shown in

Figure 4-2. An unobtrusive beep alerted subjects to errors. The highlight could then used to resume entry at the correct point in the phrase. At the completion of a phrase, a bell-like sound accompanied the change to a new phrase.

Subjects were instructed to proceed as quickly and as accurately as possible, and to try to keep their error rate to between two and five errors per phrase. Phrases were selected randomly from a bank of approximately 500, gathered from various Internet sites specializing in lists of quotations and clichés. The average phrase length was 13.9 words. A subset of these are included in (Appendix B, p. 116). The software kept track of the phrases previously entered, so that subjects were always shown unfamiliar phrases.[22]

[22] During the first few sessions, this feature of the software did not work, so that some subjects entered several phrases more than once. In general, however, phrases entered were seen only once.

Subjects were instructed to rest a while between conditions. The second condition proceeded identically to the first, except for the interface differences discussed earlier (see p. 56).

At the end of the last session, an interview was conducted with each subject. The same questions (see Appendix B, p. 118) were asked of all the subjects, however, the interview was largely open-ended, providing an opportunity to learn subjects' impressions of the FOCL method. A transcript of one subject's interview appears in Appendix D, on p. 120.

4.1.4 Design

The experiment used a within-subjects, counterbalanced design. Each subject attended 10 sessions over a period of 8 to 10 days. Sessions were spaced by no less than one hour and by no more than two days.[23] Typically, subjects would schedule one or two future sessions at the end of a session.

The independent variable was *layout type*, which could be fixed QWERTY or FOCL. The dependent variables were *text entry speed* and *error rate*. Additional data collected in the pre-experiment questionnaire for analysis as possible intervening variables included the following:

- Sex
- Age
- Typing skill (whether the subject was a touch typist or not)
- First language (English or other)
- Video game use (whether the subject regularly played video games)
- Handedness
- Number keypad skill (whether the subject could use the keypad to enter numeric data quickly and without looking)
- Discipline type (responses to the question "What is your field of study or type of work?" were coded as either 1) science or 2) arts & humanities)

---

[23] Typically, subjects attended one session per day.

- Education level

A brief explanation of why some of these items were collected is in order. Typing skill was thought to be a good way of distinguishing subjects with strong familiarity with the QWERTY layout from those with poor to moderate familiarity. Number keypad skill and video game use was collected to control for a high degree of skill in the arrow keys input method used in the experiment. Area of study or field of work was collected to control for a possible effect of cognitive style. Whether or not English was a subject's first language was collected to control for the possible negative effect of less English proficiency.

Time stamp data were collected on every keystroke to investigate the effect of chunking (see p. 28) and thereby test the second hypothesis (see p. 54). In addition, the time stamp data allowed investigation of visual search behaviour within the FOCL condition.

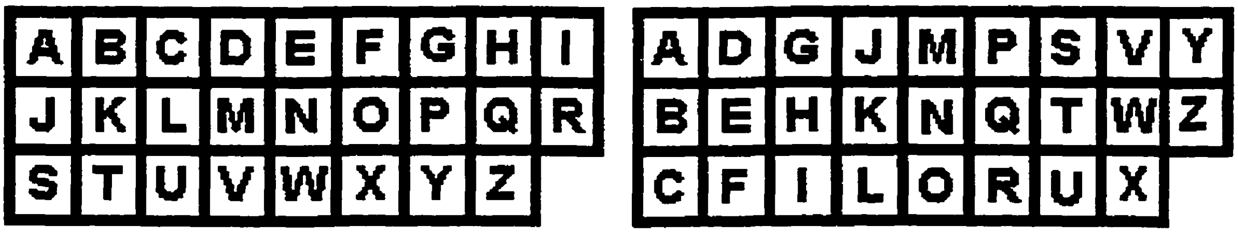


Figure 4-3: Alphabetic fixed layouts rejected for fixed condition layout.

There were several layouts considered for the fixed condition, in particular, various alphabetic layouts, such as the ones shown in Figure 4-3. While the choice of one of these would have eliminated the difference in row lengths between the two conditions, the QWERTY layout was chosen for its reputation as the standard in typing and the most familiar of layouts. This, it was felt, would increase the validity of a positive result and not leave the study open to the criticism of having compared FOCL to a weak "opponent".

4.2 Results

Of primary interest in analyzing the results of the experiment were performance differences between the two layout conditions, as observed in entry speeds and error rates, which would confirm or deny hypothesis 1 (see p. 54). In addition to comparing mean entry speeds and mean error rates for the two conditions, the sample was subdivided according to data collected in the pre-experiment questionnaire, as described on p. 59.

As all subjects were right-handed, the performance effect, if any, of *handedness* could not be measured. There was insufficient variation among the subjects in *age* and *education level* , so these were excluded from the analysis. Thus, only *sex*, *typing skill*, *first language*, *video game use*, *number keypad skill* and *discipline type* were included.

4.2.1 Entry Speed

Figure 4-4 plots mean entry speed for all subjects over the ten sessions, by layout condition.[24] Hypothesis 1 predicted that entry speeds in the FOCL condition would exceed those in the fixed QWERTY condition after a moderate amount of practice. The results force us to reject this hypothesis. Although the graph suggests slightly better performance occurred in the fixed QWERTY condition, the difference was not found to be significant ($F_{1,10} = 1.44$, $p > .05$). There was, however, a significant interaction effect of layout condition × session ($F_{9,90} = 2.62$, $p < .05$). This can be seen in Figure 4-4 as non-parallel lines between pairs of points in the two conditions, for example from session 5 to 6. As expected, there was a significant effect of time (session) on entry speed ($F_{9,90} = 178.3$, $p < .0001$). Subjects improved in both conditions throughout the experiment.

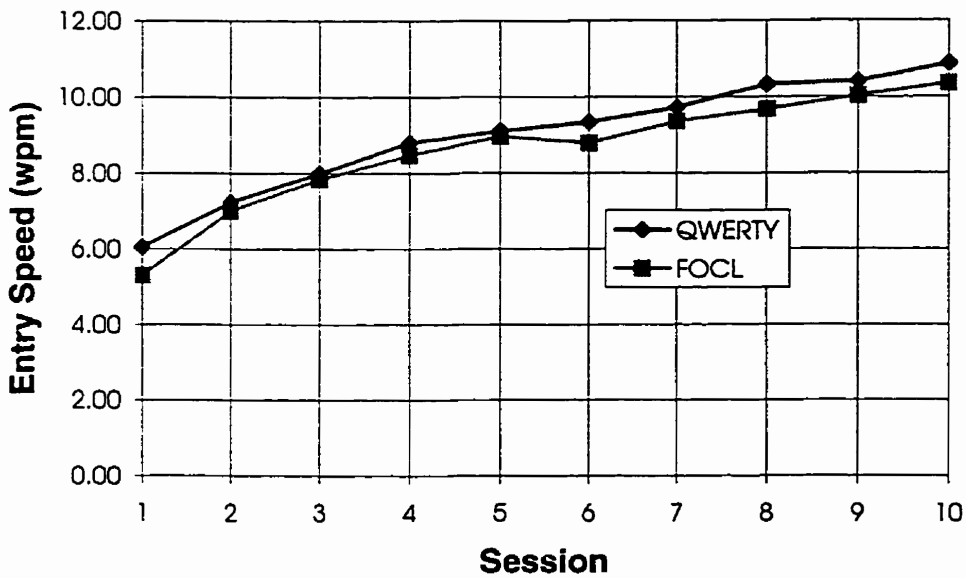


Figure 4-4: Mean entry speed (wpm) by session and layout type.

[24] The data are tabulated in Table 1, Appendix C.

Splitting the sample according to each of the secondary variables of sex, first language, etc. revealed mean differences in every case. The following differences were observed:

- Males were faster than females in both conditions.
- Native English speakers were faster than non-native English speakers.
- Regular video game players were faster than infrequent or non-players.
- Typists were faster than non-typists.
- Science types were faster than arts & humanities types.
- Proficient number keypad typists were faster than unskilled ones.

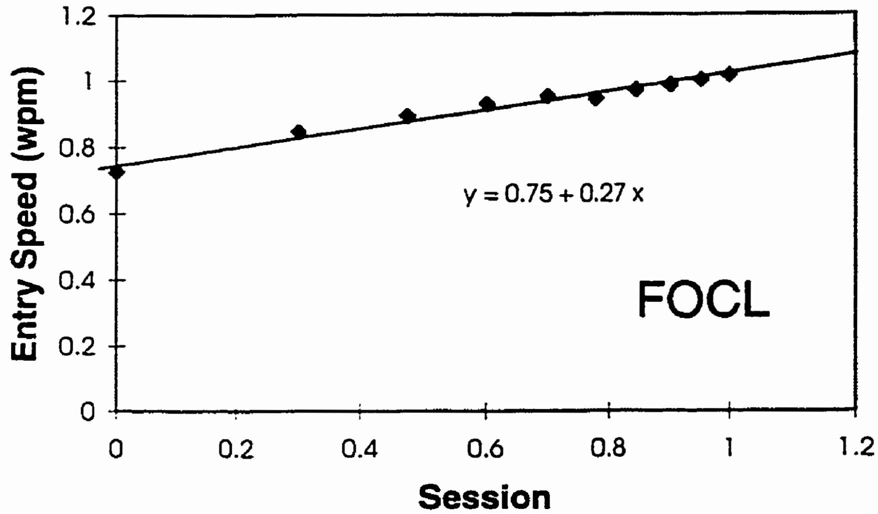
However, none of these effects achieved significance. It is conceivable that significant differences would be observed in a larger sample. Graphs of entry speed by each of these variables are included in Appendix C, p. 124.

Learning Model Fit

Table 4-1: Learning Models

Condition	Initial entry speed (wpm)	α	Learning Curve Equation	R^2
FOCL	5.31	0.27	$T_N = 5.31 \cdot N^{0.27}$	0.97
QWERTY	6.04	0.25	$T_N = 6.04 \cdot N^{0.25}$	0.99

Table 4-1 shows the specific derivations of the learning model (see p. 26) to each of the two layout conditions. Figure 4-5 shows mean entry speed by session converted to log-log coordinates, with corresponding regression lines. Recall that the α constant in the learning curve equation is the slope of this regression line. The two conditions are graphed separately because they would otherwise be too close together to view clearly. In both cases, the performance data are very close to the power law predictions, as the high correlation coefficients (R^2) in Table **4-1** attest.



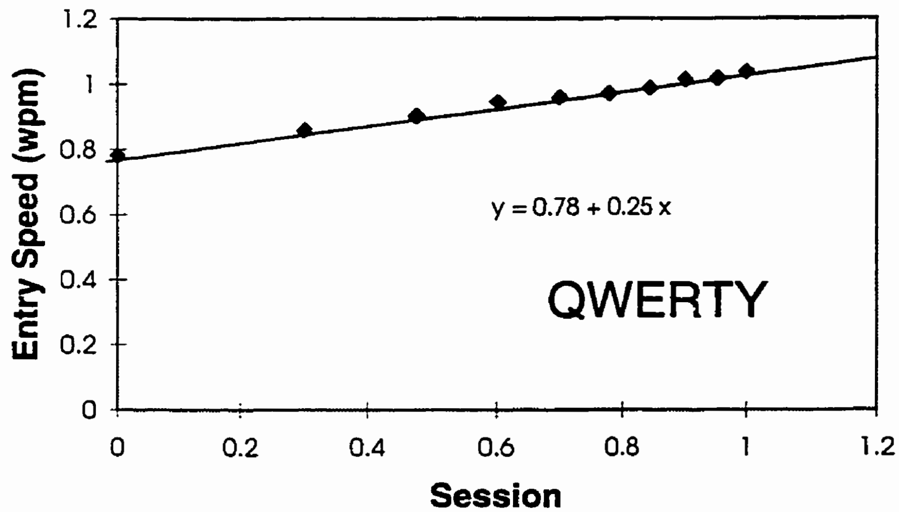


Figure 4-5: Entry speed by session in log-log coordinates, with corresponding regression line and equation.

4.2.2 Error Rate

There was a significant effect of layout condition on error rate observed ($F_{1,10} = 6.41$, $p < .05$). The effect is noticeable in Table 4-2, in the consistently lower error rates in the FOCL condition. There was not a significant effect of time (session) on error rate. As can be seen in Table 4-2, the error rates did not change drastically over the course of the experiment.[25]

Table 4-2: Mean error rate (%)by layout type

	Layout Type	
Session	QWERTY	FOCL
2	2.16	2.04
3	2.31	1.39
4	1.87	1.56
5	1.83	1.42
6	1.98	1.75
7	2.11	1.32
8	2.10	1.74
9	2.12	1.05
10	1.95	1.68
Means	2.05	1.55
Stdev	0.15	0.29

Splitting the sample according to secondary variables (sex, typing skill, etc.) revealed mean differences in all cases. The following differences were observed:

- Males made fewer errors than females in both conditions.
- Native English speakers made fewer errors than non-native speakers in both conditions.
- Typists made fewer errors than non-typists in both conditions.
- Regular video-game players made fewer errors than infrequent or non-players.
- Non-science types made fewer errors than science types.
- Unskilled number keypad users made fewer errors than proficient users.

However, as with entry speed, none of the effects achieved significance. These data are included in Appendix C, p. 124.

[25] Due to a software problem during the experiment, error data for the first session is not available.

4.2.3 Chunking

The second hypothesis states that higher FOCL entry speeds will be partially explained by chunking (see p. 28). To test this hypothesis, the time stamp data were analyzed for the presence of rapid keystroke sequences. A computer program extracted keystroke sequences in which no adjacent pair were spaced in time by more than a threshold value, t. If the N-gram corresponding to a keystroke sequence satisfying this condition was at least two characters long, the program would store the N-gram, later ouputting it along with its frequency of occurrence. Output of this sort was produced for sessions 5 through 10, for each subject individually.

After some investigation of sample output, the threshold was varied between 0.2 s and 0.4 s, producing, as one would expect, increasingly more output for higher threshold values. Values lower than 0.2 s yielded too sparse a set of output, while values higher than 0.4 produced more noise than valid output. This is to be expected, as the concept of chunking is rendered somewhat sterile with inter-keystroke rates of less than 2.5 (corresponding to 0.4 s per keystroke) keystrokes per second.

Even in the 0.2 s to 0.4 s range, the output contained a fair amount of noise, specifically, unlikely letter sequences such as KC and GT. These were attributed to user errors, which are often inadvertent, rapid key sequences, and thus indistinguishable to the program from legitimate chunks. In order to exclude such output, only N-grams detected more than once in a session were considered chunks.[26] This filtering eliminated most of the noise but excluded some legitimate chunks, mainly longer ones. Because the likelihood of multiple occurrences of the same N-gram in a session decreases for larger N-grams, the criterion was modified to exclude only *digrams* occurring just once, while N-grams *of length three or more* were included on the basis of one occurrence. The program output for threshold values of 0.2 s, 0.3 s, and 0.4 s, which is too long to display here, appears in Appendix F, p. 142. These results are discussed later in this chapter.

4.2.4 Visual Search

The third hypothesis states that the time to locate a target letter in the FOCL condition will increase with distance from the cursor. To test this hypothesis, *visual search time* was operationalized as the time between the selection of one letter and the next keystroke, and these

---

[26] Chunking implies that a keystroke pattern has been learned. The presence of more than one occurrence of an N-gram in the output for one session is a strong indication that the N-gram's associated keystroke pattern has been learned.

times were extracted from the time stamp data. This operationalization makes two simplifying assumptions. The first is that users will not strike a key (i.e., move towards a letter) until they have located it. Two counterexamples are (1) the user moving towards a letter without perceiving the new layout, as when executing a chunk, and (2) the user "hovering", that is moving back and forth, or up and down, while attempting to locate the letter. The second assumption is that visual search time accounts for all the time between selection of a letter and the next keystroke. Other sources of non-activity are decision times surrounding *route choice* (which of the possible route alternatives) and *initial direction* (which key corresponds to the direction of the first move in a route) (Welford, 1968). It might therefore be more appropriate to call the interval between selection and next keystroke *hesitation time.* For our purpose, however, it is a reasonable approximation of visual search time. A more complex model of visual search, while worthwhile, is beyond the scope of this thesis.

Hesitation times were extracted from the time stamp data and grouped according to both horizontal and vertical distance from the cursor. Means were then calculated for each distance, by session. This was done with time stamp data from sessions 6 through 10 only. Unfortunately, the time stamp collection component of the software was not operational during the first five sessions of the experiment.

Vertical Distance from Cursor

The vertical distance results are shown in

Figure 4-7. In the FOCL condition, targets in row 1, take less time to locate than targets in row 2, which take considerably less time to locate than targets in row 3.[27] This result supports the finding that scanning of alphanumeric arrays tends to proceed in a left-to-right, top-to-bottom fashion (Ford, White & Lichtenstein, 1959), as we would thus expect targets further down to take longer, on average, to locate. In the QWERTY condition, although hesitation time appears to decrease moderately with vertical distance from the cursor, the variance is quite low, i.e., hesitation time is essentially stable, as we would expect. In both conditions there is a gradual decrease in hesitation time between sessions 6 and 10, as expected. Practice reduces reaction time (Welford, 1968).

---

[27] We can refer to row numbers because of the fact that the path to a letter in the FOCL condition always begins at the top-left corner (i.e., the cursor home position). Targets that are a vertical distance of 0 from the cursor are in row 1, a distance of 1 in row 2, and so on.

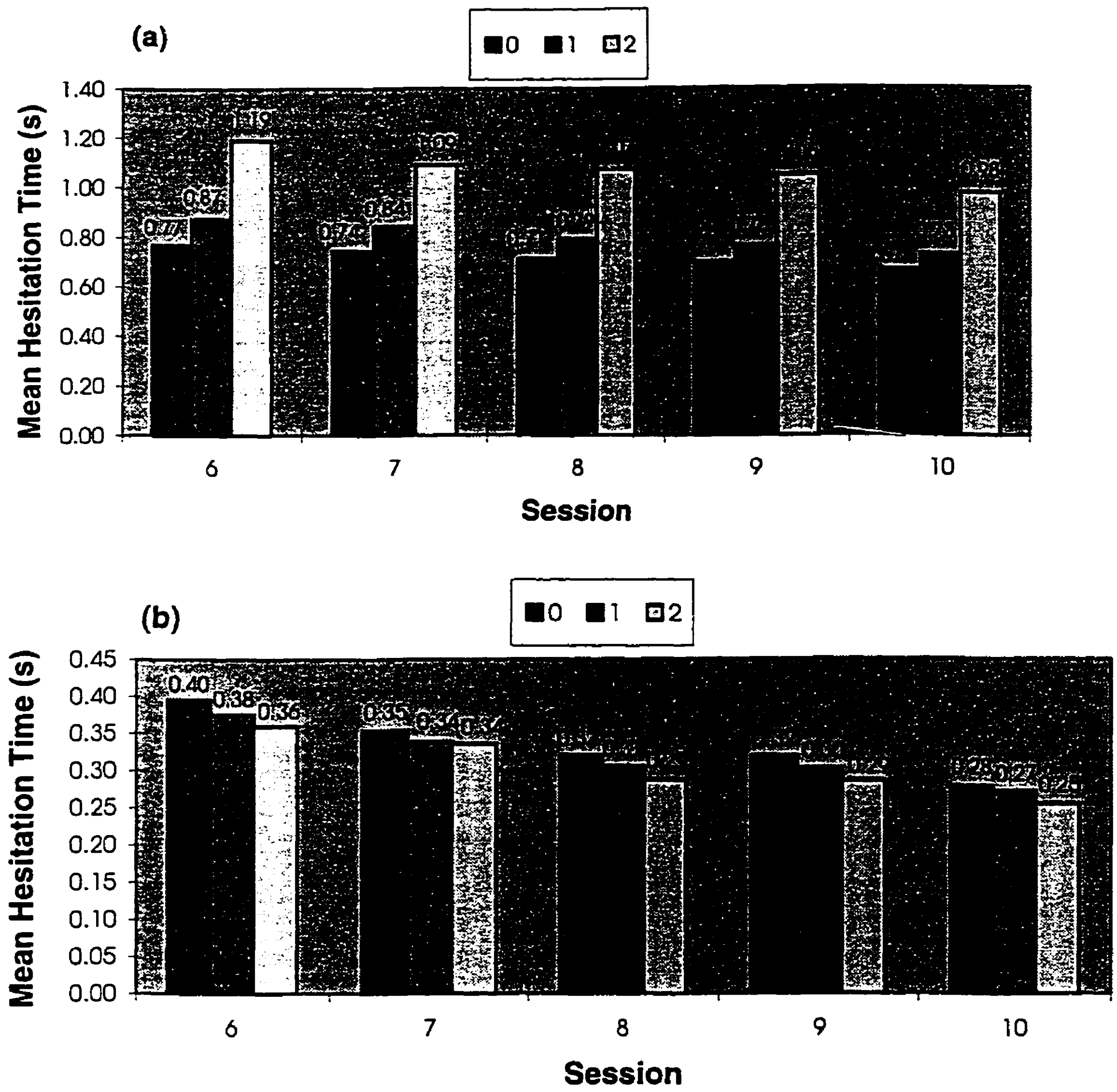


Figure 4-7: Mean hesitation time (s) by session and vertical distance (keys) from cursor for (a) FOCL, and (b) QWERTY conditions.

Horizontal Distance from Cursor

The horizontal distance results for the FOCL and QWERTY conditions are shown in Figure 4-8. Only times up to a distance of five horizontal positions are included in the graph of the FOCL condition due to the extremely low number of target occurrences beyond that distance. There is a noticeable effect of horizontal distance from cursor on hesitation time in the FOCL condition, although the effect is not as unidirectional as that of vertical distance. Curiously, targets in column 1 (corresponding to a horizontal distance of 0) take consistently longer to locate than targets in column 2.

In the QWERTY condition, hesitation times are more stable, as with the vertical distance data. The data from session 8 most clearly demonstrate this stability, which is an expected result, given the assumption of familiarity with the QWERTY layout.

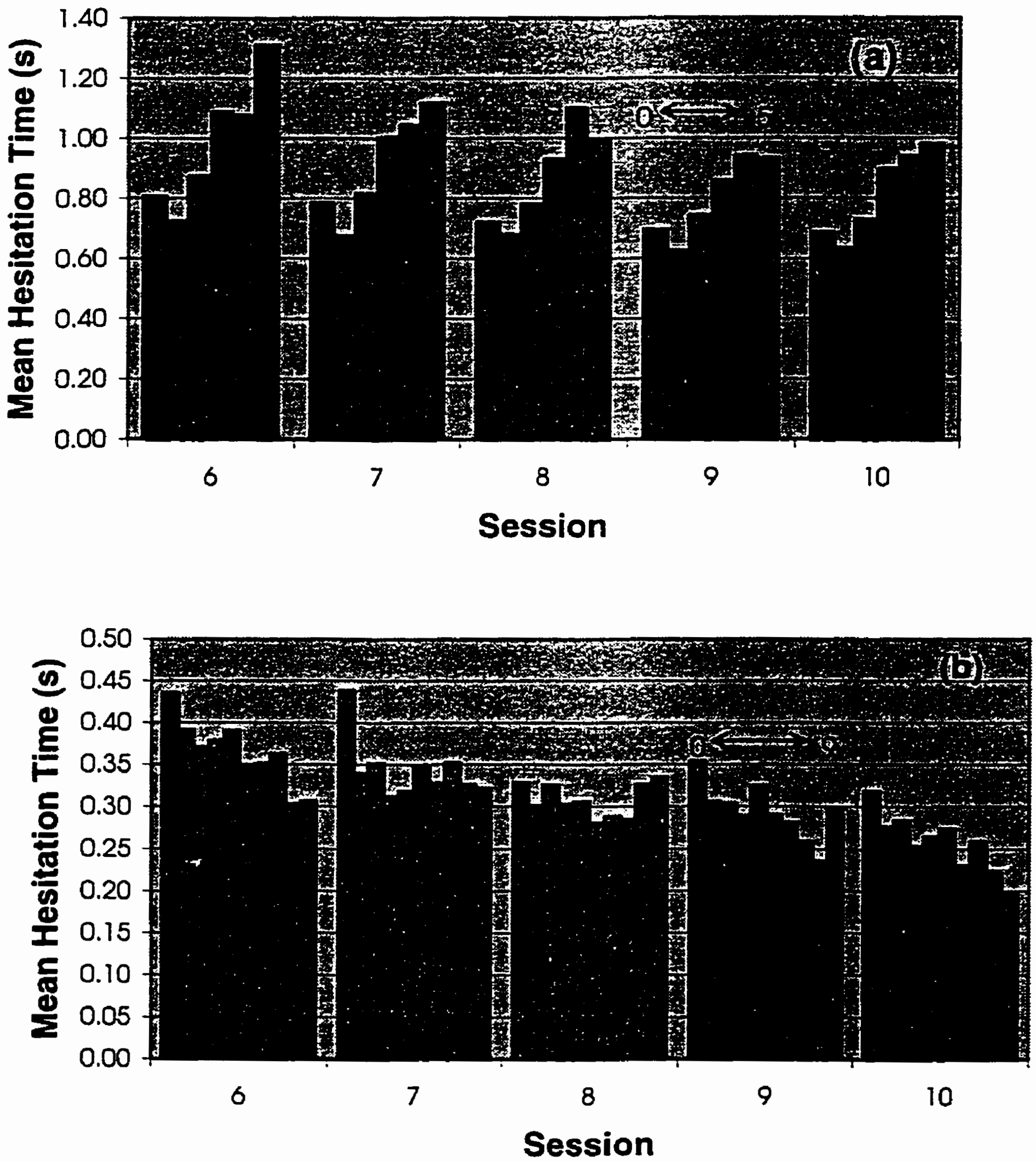


Figure 4-8: Mean hesitation time (s) by session and horizontal distance (keys) from cursor for (a) FOCL, and (b) QWERTY condition.

4.2.5 Interview Responses

Visual Scanning Pattern

Though for the most part subjects understood the statistical implications of how characters were positioned in the FOCL layout, none of them reported using optimal or near-optimal scanning patterns. Self-reported scanning patterns included left-to-right by row, left-to-right by groups of 2-3 columns, and random.

Frustrating Aspects of the FOCL Interface

Subjects expressed frustration with the behaviour of the FOCL prototype in certain situations. These included the following:

- The need to relocate a letter in the event of a double letter, rather than being able to select it twice.
- The inability to take advantage of chance adjacencies in the character set that were identical to a letter combination in the word being entered.
- The change in position of letters, particularly common ones, from one layout to the next.
- The perceived poor placement of the letters I and P in the layout following a space character.

Chunking

When asked to list words and letter sequences they had learned to enter as chunks, subjects reported a surprisingly small number. The words THE and YOU figured prominently, however, there was very little variety between subjects, except that one or two subjects were not aware of having chunked at all.

4.3 Discussion

Entry Speed and Error Rate

Although the first hypothesis, predicting higher FOCL entry speeds, must be rejected, the fact that the FOCL method did not perform significantly worse than the fixed QWERTY method is encouraging. It suggests that even a moderate improvement to the FOCL design is capable of raising performance to levels that exceed the fixed approach. Redesign possibilities are discussed in the next chapter.

The single highest FOCL session mean of 13.34 wpm, is less than 2 wpm short of the lower bound of handwriting speeds, 15 wpm (Card, et al., 1983), the benchmark of this research. The single highest QWERTY session mean of 15.66 wpm is within the handwriting range.

The observed trend of consistently lower error rates on the FOCL condition is probably a function of fewer keystrokes and thus lower error probability. The more interesting result with respect to error rates is how low they are for this interaction technique, in general.

Learning Effects

Because the FOCL log-log linear regression slope (0.27 words/min^2) is greater than the fixed QWERTY log-log linear slope (0.25 words/min^2), there is, theoretically, a crossover point (see p. 31) at which FOCL performance would exceed fixed QWERTY performance. However, the difference in slopes (0.02) is so small, that this point is, for all practical purposes, unachievable, as it would require an enormous amount of practice. According to the model, entry speeds for FOCL would begin to exceed those of QWERTY after 640 sessions, or roughly 160 hours or practice!

A more realistic question to ask is the amount of practice required for FOCL to match handwriting speeds, that is 15-17 wpm (Card, et al., 1983). According to the model, speeds of 15 wpm would be achieved after 47 sessions, or roughly 11.75 hours, and speeds of 17 wpm would be achieved after 75 sessions, or roughly 18.5 hours. This clearly rules out the technique, in this first instantiation, as a viable mobile text entry method. Designs that achieve handwriting speeds at a faster rate must be sought.

Chunking

As chunking is defined in terms of the chunking theory (Newell & Rosenbloom, 1981), it is necessary to state what constitutes support for the theory. Table 4-3 lists the theory's predictions and defines observations in the data that would confirm them.

Table 4-3: Chunking theory predictions and observations within FOCL chunk data that would lend support to the predictions.

	Chunking theory prediction	Supporting observations
(a)	Shorter chunks will be acquired before larger ones.	• Presence of exclusively small chunks, i.e., digrams, in early sessions • Presence of exclusively smaller chunks at lower threshold values for the same session
(b)	The larger a chunk, the longer it will take to acquire.	• Presence of longer chunks, i.e., $N \geq 3$, at later sessions only • Presence of longer chunks at higher, but not lower, threshold values for the same session
(c)	Larger chunks will be acquired as a result of combining smaller ones.	• First appearance of a long chunk at some session, and presence of smaller component chunks in prior sessions, e.g., OU -> YOU -> YOUR • Appearance of a long chunk at high threshold value, and presence of smaller component chunks at lower threshold values

It is unfortunate that data for sessions 1 through 4 are unavailable. The available data do not capture novice skill level, and so the chances of finding strong evidence of chunking are reduced.[28] Nonetheless, some support for the theory is evident in the data as is. The reader is asked to consult Appendix F when reading the following examples of support for the theory.

Support For (a)

Subject 4, $t = 0.2$ s: The only chunks in sessions 5 through 9 are TH and HE.

Subject 8, $t = 0.4$ s: Chunks in sessions 5 through 9 are exclusively two-letter

[28] Even if data for all ten sessions were available, chunking might still not be observed. Chunking may take many hours of practice to surface (Crossman, 1959; Newell & Rosenbloom, 1981, Welford, 1968) and be detectable using the methods described.

Support For (b)

Subject 4, $t = 0.2$ s: The only chunks in sessions 5 through 9 are TH and HE.

Subject 8, $t = 0.4$ s: Chunks in sessions 5 through 9 are exclusively two-letter

Support For (c)

Subject 4, $t = 0.2$ s: THE, which incorporates the earlier appearing TH and HE, first appears as a chunk in session 10.

Subject 4, $t = 0.3$ s: OU appears in sessions 5 and 7, while OUT appears first in session 8. Also, NG appears in session 9, while the first appearance of ING occurs in session 10.

Subject 6, $t = 0.3$ s: OU appears in session 5, while YOU first appears in session 6.

Subject 6, $t = 0.4$ s: TH, HA, and THA appear as chunks before the first appearance of THAT and WHA in session 10.

The set of observed chunks is rather limited. A glance at the data in Appendix F reveal a predominance of TH, HE, and OU, and *N*-grams or words that incorporate these chunks, such as THE and YOU. As these correspond to simple key sequences[29] in the FOCL prototype, their profusion is not surprising. What is unexpected is the absence of chunks corresponding to more complex keying sequences. This finding is in agreement, however, with the small lists of chunks reported by all subjects in the post-experiment interviews. If nothing else, this agreement partially validates the approach used to detect chunks.

The data suggest two kinds of users with respect to chunking: those that do and those that do not. Subjects 1, 2, 5, 7 and 10 exhibit little or no chunking behaviour. Some chunking begins to emerge at $t = 0.4$ s with subjects 5 and 10, suggesting that perhaps these users are slower learners than the others, i.e., that individual differences may explain the lack of chunking in some subjects. However, there is clearly a group of non-chunkers, distinct from slow learners. Subjects 2 and 8 offer the most striking examples: the word THE does not appear in any of their output. A possible explanation for this behaviour is that these subjects did not heed the instruction to avoid reading phrases character by character. This would also explain the near-exclusive presence of digrams in the chunks that emerge with these subjects at $t = 0.4$ s.

[29] Sequences involving the selection key exclusively after the first character of the *N*-gram is entered.

While the repertoire of chunks is small--at least within the scope of the experiment--there are examples in the data of increasing frequency of acquired chunks with practice. Without knowing frequency of each chunk in a session's phrases, however, it is impossible to confirm or deny this. To explore this further, the frequencies of THE, YOU, and AND were computed for the phrases entered by subject 5. At $t = 0.3$s, Figure 4-9 shows the percentage of occurrences of each word chunked from session 5 through session 10.

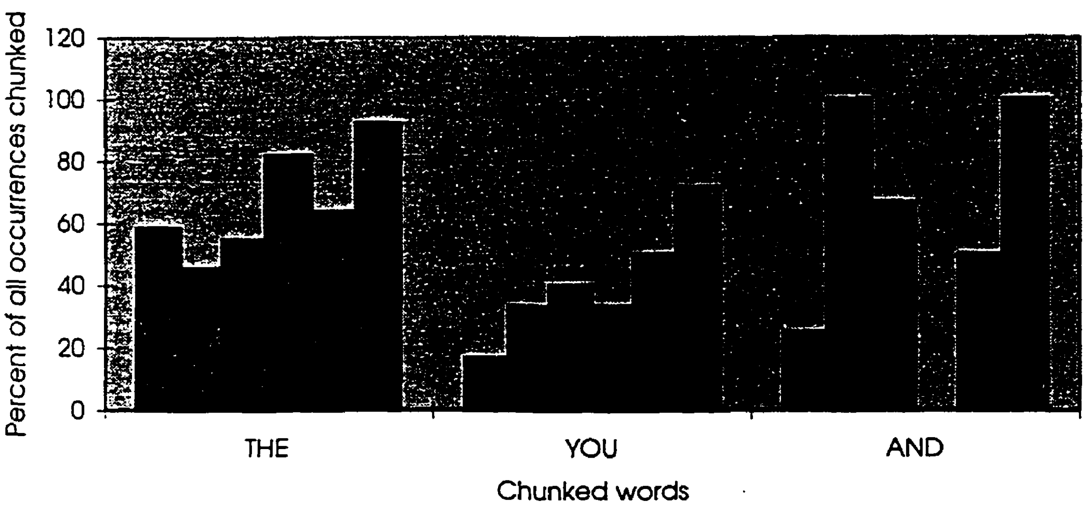


Figure 4-9: Percentage of occurrences of the words THE, YOU, and AND chunked by subject 5 during sessions 5 through 10, assuming an inter-keystroke threshhold of 0.3 s. Session 5 is the leftmost bar in each group of bars.

The graph shows that as the experiment progressed, this subject increasingly entered THE and YOU as chunks when these *N*-grams were encountered. The pattern is inconclusive in the case of AND, but this may be due to the low frequency of that word–in the sessions in question, it occurred only twice per session, on average. While a more elaborate analysis of the data is required to confirm the trend, the graph suggests a possible relationship between practice and the likelihood a chunk will be executed when its associated *N*-gram is encountered, which is a predicted outcome of the chunking theory (Newell & Rosenbloom, 1981). This is also reminiscent of the concept of *strength* of a response or production rule (Anderson, 1982).

Chunking did not impact the FOCL condition as anticipated. Although the data show that chunking does occur, there is little support for the second hypothesis, which predicts that chunking will be a factor in the improvement in entry speeds in the FOCL condition. The data used to illustrate increasing frequency of acquired chunks (see Figure 4-9, p. 73) represents the best, not the average, example of chunking in the subject sample. Subjects who displayed little or no chunking behaviour nonetheless improved their entry speeds in the FOCL condition over the course of the experiment. Whether those subjects who displayed chunking behaviour performed significantly better than those who did not is a question for further investigation.

Chunking appears to be limited to *N*-grams associated with simple keystroke sequences. Skill continues to improve indefinitely (Crossman, 1959; Welford, 1968), so it is likely that more complex chunking would take much longer to surface. However, unlike in touch-typing, where a word of length *n* always requires *n* keystrokes to enter[30], in the FOCL method, *N*-grams of the same length may vary greatly in the number of keystrokes required to enter them. The limiting factor in typing is the awkwardness of the keystroke sequence, not its length. Several of the skill acquisition theories (Anderson, 1982; Crossman, 1959; Newell & Rosenbloom, 1981) predict that longer response patterns will take longer to acquire, so it is perhaps not surprising that we do not observe the emergence of more complex chunks in the FOCL condition.

Visual Search

As fully expected, hesitation times are, on average, longer in the FOCL condition. A large proportion of this difference must be attributable to visual search, since that is the primary difference between the tasks in the two conditions. The third hypothesis, which predicts a positive relationship between visual search time and distance from the cursor in the FOCL condition, and no relationship between visual search time and distance from the cursor in the fixed QWERTY condition, is supported by the results. On average, hesitation time for targets in the FOCL condition increases both with vertical and horizontal distance from the cursor. On the other hand, hesitation time is relatively stable in the fixed QWERTY condition, regardless of distance from the cursor. Insofar as hesitation time is an acceptable approximation of visual search time, there is support for the claim of hypothesis 3 that targets in the FOCL condition take longer to locate the further they are from the cursor, whereas all targets in the fixed QWERTY condition take the same amount of time to locate. Though the validity of using hesitation time to approximate visual search time is open to question, the range of values reported are consistent with the mean reaction time predicted by the Hick-Hyman law of choice reaction time (Hick, 1952; Hyman, 1953; Welford, 1968), which states that the average time to locate a target in a display, $\overline{RT}$, is proportional to the logarithm of the number of alternatives:

$$\overline{RT} = 0.200 \log_2(N) \tag{4}$$

[30] Assuming no change of letter case.

For $N = 26$[31], the equation predicts an average search time of 0.924 s, which is in the middle of the range of hesitation times observed.

Given the widespread familiarity with the QWERTY layout, we would expect there to be little or no difference in search times from session 6 to session 10. However, the hesitation times in the QWERTY condition drop over time by as much as those in the FOCL condition, as visible in Figure 4-7 and Figure 4-8. This is due to an increase in response rate to primitive patterns that is a commonly observed aspect of skill acquisition (Keele, 1986).

The higher hesitation times for targets a horizontal distance of zero from the cursor (i.e., including targets at the cursor position) than those a distance of one is surprising. However, subjects were observed on a number of occasions failing to spot the target at the cursor position itself. This could explain the higher hesitation times. Perhaps something about the way the cursor was highlighted (it was inverted) makes the letter in that position less noticeable before cursor movement has drawn the eye's attention to it. This should be investigated further.

In the post-experiment interviews, several subjects stated that they could quickly process a 3 by 3 area of letters and determine whether the target letter was there. If it was not, they would begin to search from left to right. During the experiment subjects were observed on numerous occasions having difficulty finding a letter on the right side of the layout. The increase in average hesitation time with horizontal distance from the cursor, combined with these observations, suggests that FOCL suffers badly when target letters are further away from the cursor. Since overall FOCL entry speeds are not significantly worse than QWERTY speeds, we can conclude that FOCL outperforms QWERTY when letters are close to the cursor, and thus more easily located. The next design phase should therefore focus on determining at what point the fluctuating approach ceases to be advantageous.

Scanning patterns

The character positioning principle of FOCL is fundamentally incompatible with the way in which people typically scan a display. Even though users can be told that a certain behaviour is optimal, they may not want, or be able, to perform it (Carroll & Rosson, 1987). In this case, the latter is likely the case, as the reader will have noticed when attempting to scan the layout in Figure 3-3 on p. 38. This incompatibility has significant redesign implications, as will be discussed in the next chapter.

---

[31] We exclude the space character, which would make $N = 27$, because its special treatment in the interface does not require that it be searched for.

Chapter 5: The Redesign Process

Although the negative result of the first experiment was unexpected, it provided a rich set of information for improving the FOCL design. A number of redesign strategies were considered, but only the one outlined in this chapter was pursued, due to time and resource constraints. Two other redesign strategies are discussed in Chapter 7.

5.1 FOCL's Point of Diminishing Returns

The critical realization from the results of the first experiment is that the benefit of the FOCL technique--fewer keystrokes--is realized when the target letter is close to the cursor, and thus easily located. The technique often suffers badly when the target letter is further away from the cursor, frequently requiring very long search times.

Although letters frequently occur close to the cursor[32], the occasional long searches for distant targets are devastating to overall entry speed. Thus, in experiment 1, much fewer keystrokes were observed in the FOCL condition than in the QWERTY condition, but no better performance. With the experiment 1 prototype, the benefits of reduced keystrokes clearly *do not* outweigh the cost of visual search time.

The realization that, for letters at some distance from the cursor, FOCL performs badly, on the whole, led to a more detailed examination of the probability data used to generate layouts. The goal was to determine what proportion of text entry is 'covered' by all the letters that are positioned in a given slot, or set of slots, in the layout. For example, if only one position in the layout were fluctuated, what proportion of the time would that position contain the target letter?

We can estimate how often a target letter will occur in n fluctuating positions in a FOCL layout by cumulatively summing the probabilities associated with the digrams in each column of the alphabetic sequence table (Table 2-4, p. 23). Recall that the rows of this table are alphabetic orderings sorted by decreasing probability to follow each letter. Thus, each column represents 27 of the 27^2 possible digrams. As Figure 5-1 illustrates, if $n = 1$, the probability of the target letter appearing in the single fluctuating position is .29. If $n = 3$, the target letter has a probability of

.53 of occurring in one of the three fluctuating positions (i.e., over 50% of the time). And if n = 10, the probability of the target letter appearing in a fluctuating position is .88 (i.e., almost 90% of the time).

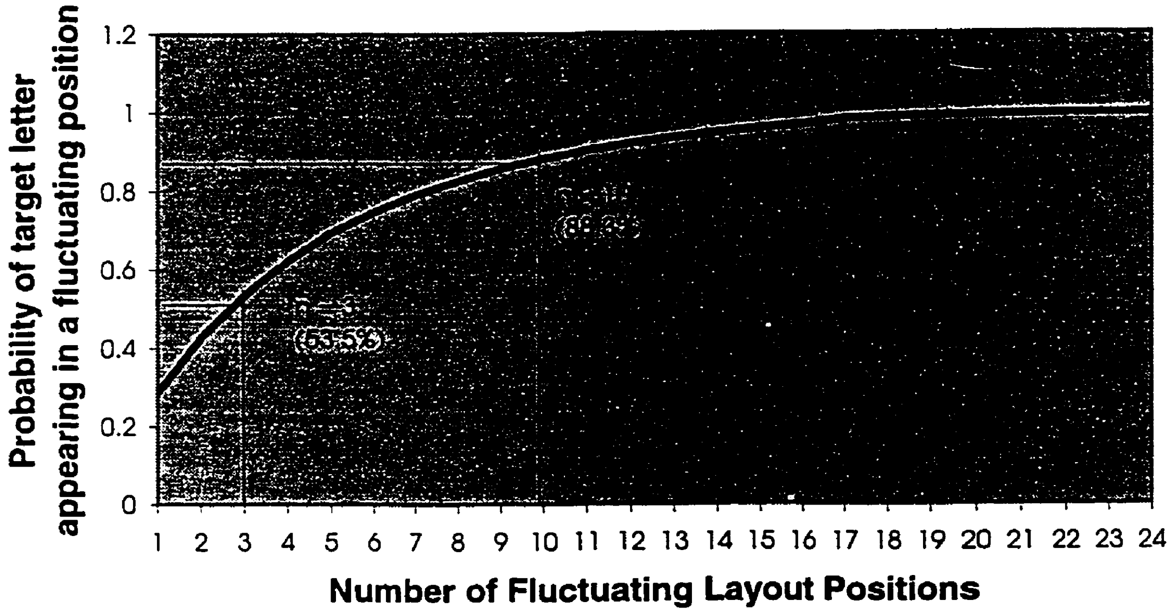


Figure 5-1: Probability a character will appear in one of the fluctuating positions of a hybrid layout, as a function of the number of fluctuating layout positions. The underlying statistics are the digram frequencies of Table 2-3, p. 22.

Two important conclusions can be drawn from this finding:

- *There is no need to have the entire layout fluctuate.* In fact, it is probably detrimental to do so. The additional time required to search more positions in the layout probably outweighs any benefit in reduced keystrokes.

- *There is very little benefit to be derived from more than 10 fluctuating positions.* The remaining 16 possible positions could only contain the target letter about 12% of the time.

Although this finding was initially surprising, it should not be. This is a reflection of the redundancy of language discussed in section 2.2.1 (p. 11). Since the distribution of digrams in English is highly non-uniform, the fact that 10 columns, or 270 digrams, cover 90% of English, should come as no surprise, given that the columns are sorted by decreasing probability.

[32] By virtue of the statistical structure of English described in Section 2.2.1.

The point is further illustrated if we examine the three figures on the next page. In Figure 5-2 we see that at about column 10, the average probability of a column's 27 digrams is less than 0.001. Figure 5-3 shows that the maximum digram probability in a column plateaus at about 0.007 at column 12, with only a minute decrease until column 18. Finally, Figure 5-4 shows the percentage of digrams in each column whose probability is zero. By column 16, half the digrams in a column do not occur at all.[33] Clearly, there is a point of diminishing returns as the number of fluctuating characters in a layout increases.

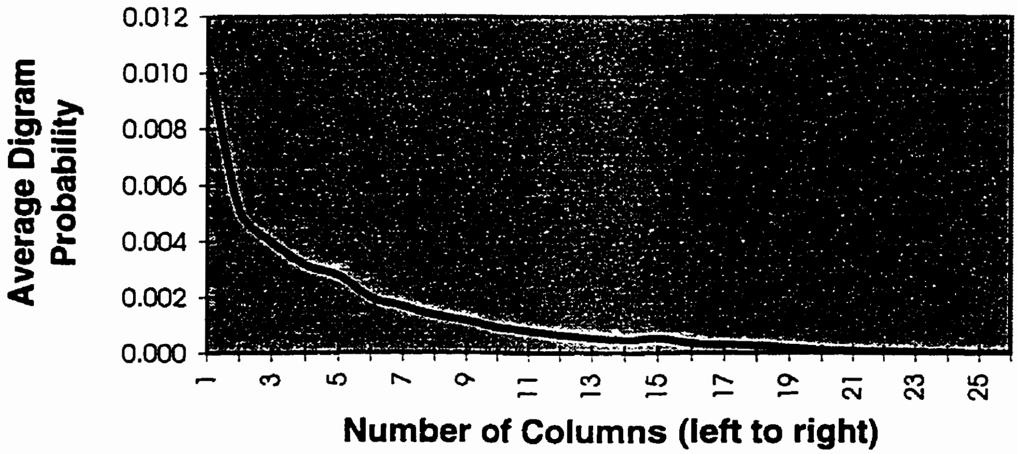


Figure 5-2: Average digram probability by column of Table 2-4

[33] This statement should be qualified: These digrams do not occur in the sample used. They may occur in other samples of English. Because of the representativeness of the sample, however, we can be extremely confident that these are highly improbable digrams.

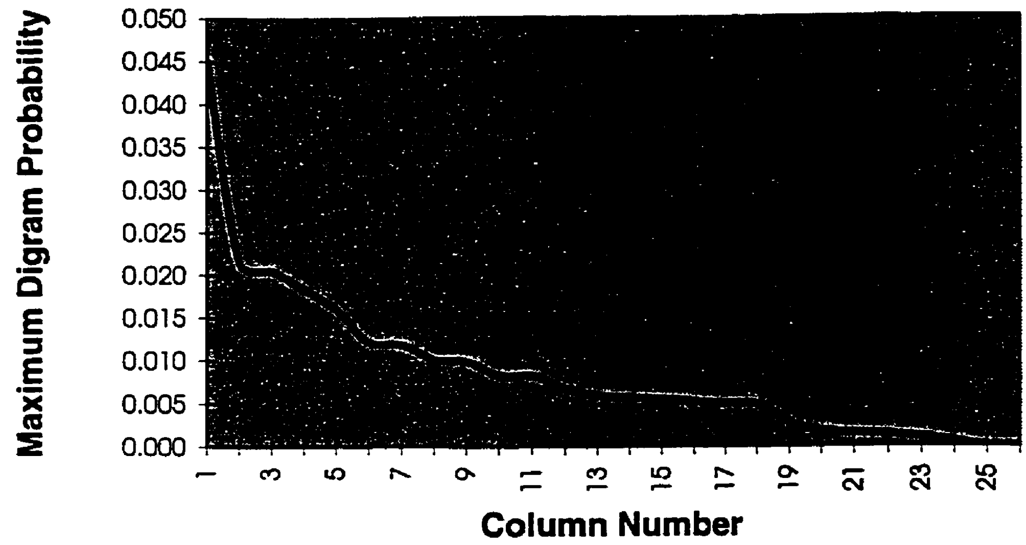


Figure 5-3: Max. digram probability by column of Table 2-4

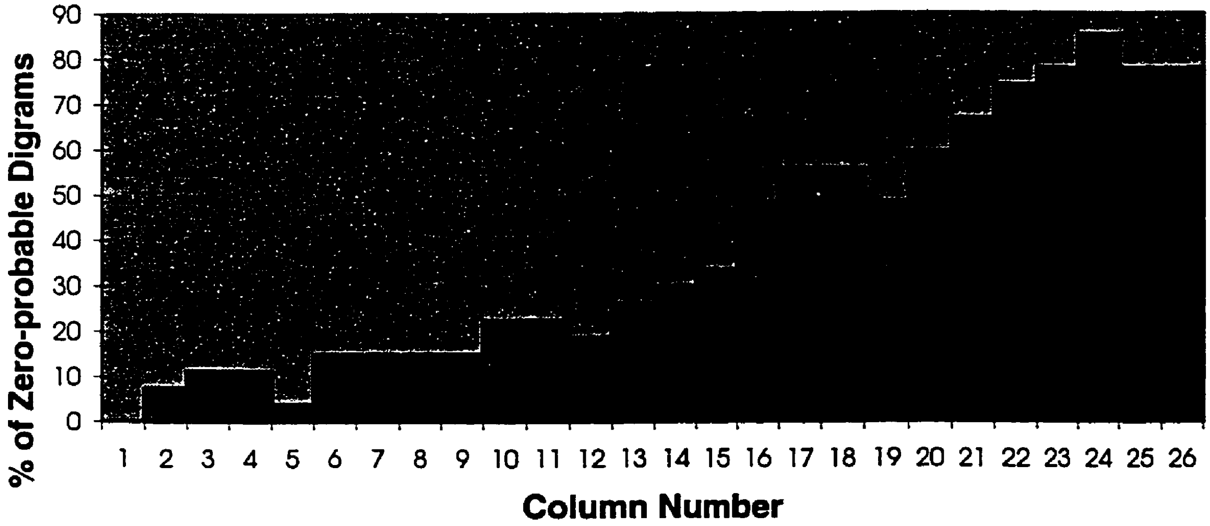


Figure 5-4: Percentage of zero-probable digrams in each column of Table 2-4 (see p. 23)

5.2 Ways of Exploiting the Point of Diminishing Returns

The point of diminishing returns has strong design implications. Two ways of exploiting it are discussed in this section.

5.2.1 Increasing FOCL's Predictive Power

If venturing beyond a certain distance from the cursor is problematic, then one solution is to reduce the occurrence of such instances. If more target letters occur in the small number of positions surrounding the cursor, then the negative aspect of FOCL will be lessened. The way to achieve this is by moving to a higher order N-gram model. Recall the example on page 15 involving the word QUIT. Higher order predictions would reduce the occurrence of such instances.

The costs of such an approach are twofold. First, the number of alphabetic orderings increases exponentially as N is increased. As it is essential that the layout updates be instantaneous, this makes increasing memory and processing demands of the system. Second, as the number of possible layouts increases exponentially, the user's ability to predict upcoming character positions is severely reduced.

5.2.2 Hybrid Layout

The other solution attempts to have the best of both the fixed and fluctuating worlds. FOCL reduces entry time for letters close to the cursor, while a fixed layout avoids the problem of long search times with letters far away from the cursor. These complementary benefits recommend a hybrid layout, a part of which fluctuates, and a part of which is fixed. The principle of usage is as follows: *if the target letter is not found in the fluctuating part, then the user can proceed directly to its position in the fixed part.*

In keeping with the goal of increasing complexity only if no other options are available, the hybrid approach was chosen as the redesign strategy to pursue.

5.3 Hybrid Design Issues

A number of new design issues emerge from the notion of a hybrid layout. These are divided into those that concern the fluctuating part, and those that concern the fixed part.

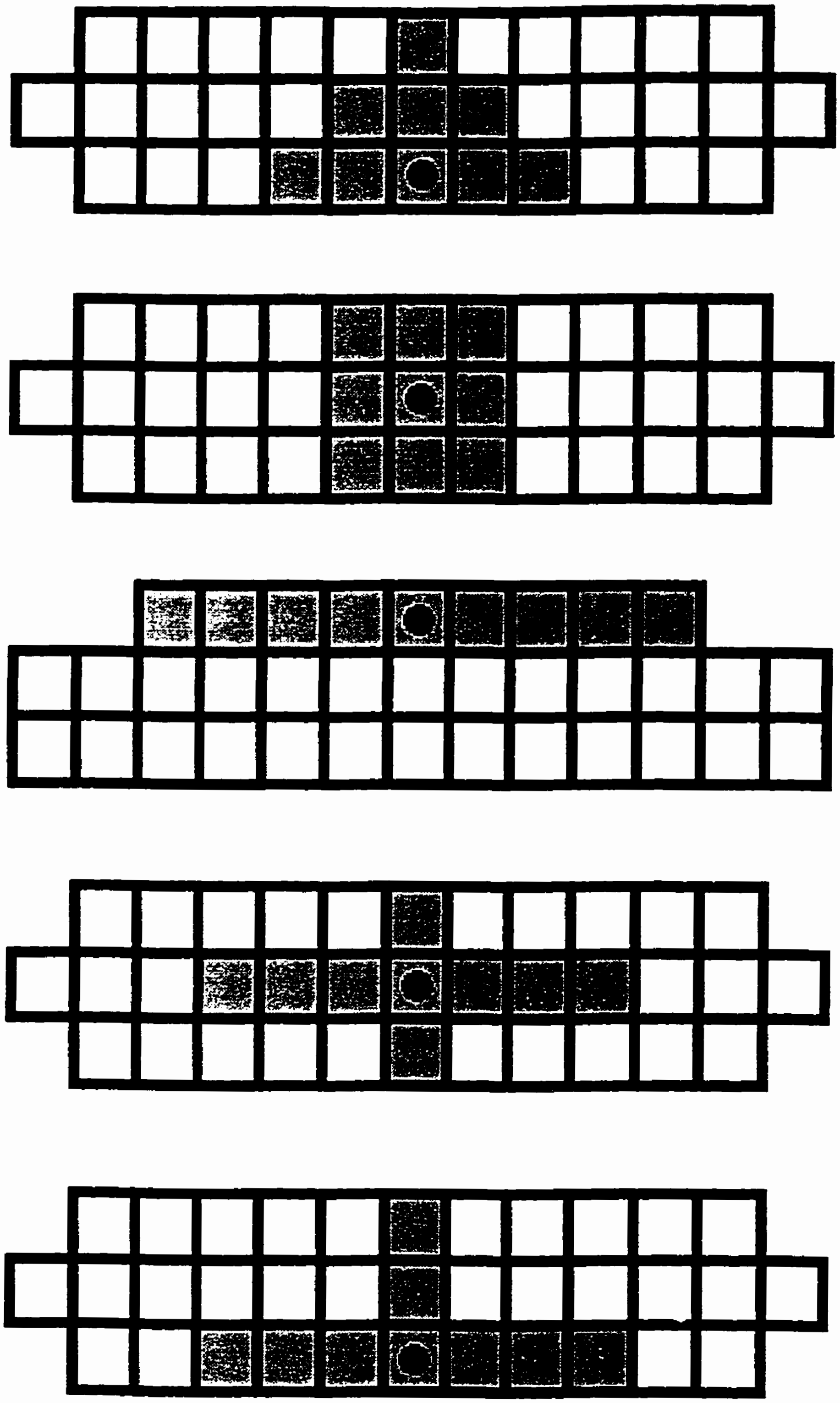


Figure 5-5: Possible arrangements of nine fluctuating positions (grey) within a hybrid layout.

5.3.1 Fluctuating Part of Layout

Size

What is the optimal number of fluctuating positions? Section 5.1, on the point of diminishing returns, provides numerous data on which to base this decision. The decision is contingent on other factors, discussed below, such as how the fluctuating positions are arranged. Also, the number of fluctuating positions may be constrained to multiples of the number of rows or columns in the layout.

Shape

How should the fluctuating positions be arranged? This depends on the size of the fluctuating part. Figure 5-5 shows several possibilities with 9 fluctuating positions. A straight line has the advantage of conforming to human scan patterns, while a cluster reduces keystrokes. The advantage of a T-shape is that every key is accessible by pressing a single arrow key--assuming the cursor home position is at the position indicated by the circle.

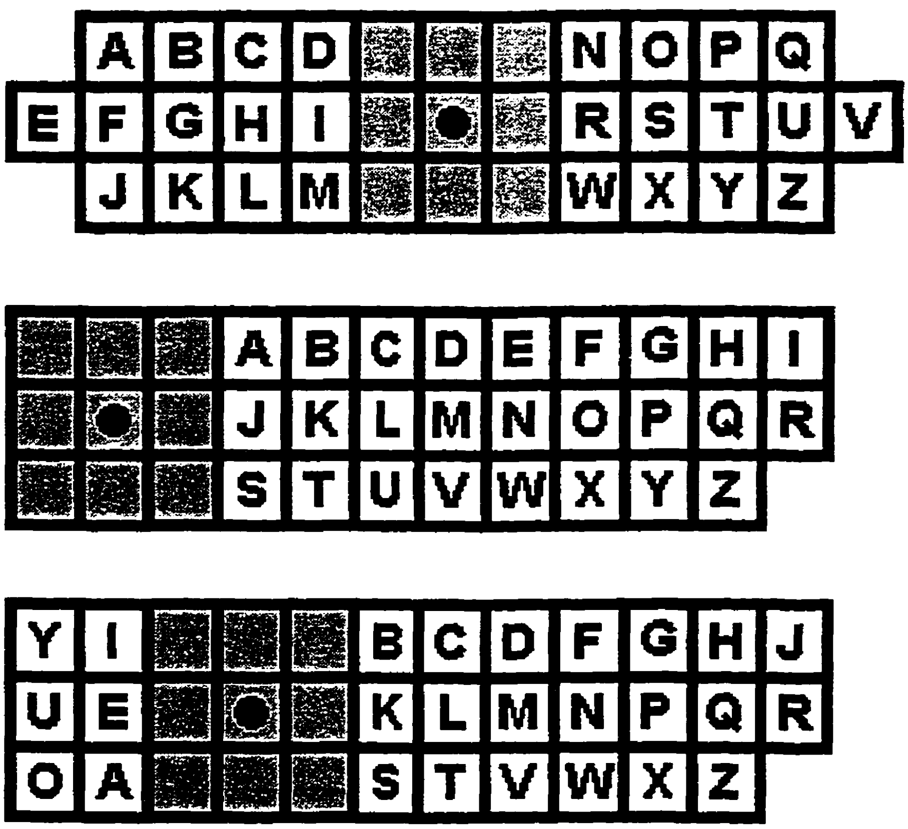


Figure 5-6: Possible placements of a 3 x 3 arrangement of nine fluctuating positions, with alphabetic fixed parts.

Position

Where should the fluctuating part be placed within the layout? Several possibilities for a 3 x 3 cluster are shown in Figure 5-6. A side placement may lead to occasional long distances to a fixed position, although this might be solved by allowing cursor-wrapping to quickly get to the other side of the layout. A centre placement reduces average distance to fixed positions, but creates an arbitrary split in the fixed part which will increase mental load as users have to determine the side of the layout in which the target letter is located.

5.3.2 Fixed Part of Layout

Layout of Characters

Depending on the size of the fluctuating part ($\overline{F}$) the layout of the fixed part becomes more or less critical. If $\overline{F}$ is large, then recall that the fixed part will be visited very infrequently. In this case, the degree to which the fixed part minimizes keystrokes is not significant. A more important consideration is learnability of the fixed layout, the goal being to eliminate the hesitation that plagued the first FOCL prototype. If $\overline{F}$ is small, on the other hand, the fixed part should be arranged as optimally as possible. This would depend on the placement of the fluctuating part, as common letters would need to be placed close to it.

Handling Redundancies

How should letters that occur in the fluctuating part be treated in the fixed part? There are three possible treatments:

- Do not display such letters in the fixed part
- Mark them in some way, such as by dimming, colour or a graphical symbol
- Do nothing

The goal is to prevent the user from unnecessarily moving to a letter in the fixed part when it is in the fluctuating part. The first treatment can be ruled out because it introduces uncertainty to the fixed part, the very problem we are trying to solve. Marking redundant letters may be beneficial, but must be done so that the mark is not distracting. Dimming may be too dramatic a change to occur with every character entered. Doing nothing may in fact be the best alternative. It makes the reasonable assumption that users will migrate towards optimal use with time.

5.4 Redesigned Prototype

5.4.1 Description

The redesigned hybrid prototype (Figure 5-7) uses a QWERTY layout for the fixed part. The fluctuating part consists of nine positions, arranged in a 3 x 3 matrix, placed at the centre of the layout. The split in the QWERTY layout corresponds exactly to the keys covered by the two hands in standard typing, a split common now in ergonomic keyboards. The cursor home position is in the centre of the fluctuating part.

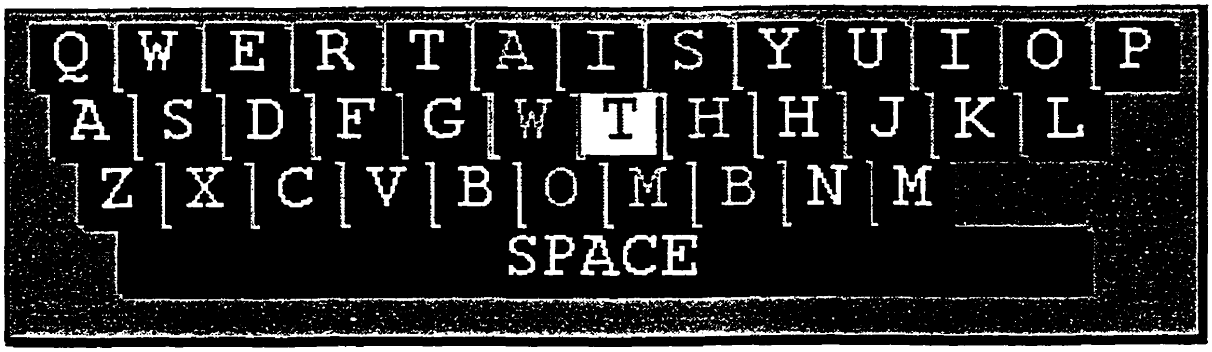


Figure 5-7: Hybrid layout design

5.4.2 Design Rationale

The choice of QWERTY as the fixed layout is an acknowledgement of it being the most familiar to the most people. This, it was reasoned, would facilitate learning of the tool by eliminating the need, for most people, to learn a new fixed layout. The fluctuating part was positioned at the centre of the layout to reduce average distance to fixed positions. The fluctuating keys were arranged as a 3 x 3 cluster because it was felt that this arrangement would afford rapid search *and* position all the fluctuating keys in close proximity to the cursor. The cursor home position was placed at the centre of the layout so that no fluctuating key would be more than two keystrokes from the cursor. While this meant that the space character would require two keystrokes to enter, rather than the one of the previous prototype, this was judged to be a reasonable tradeoff given the other benefits of a centre cursor.

Chapter 6: Second Experiment

A second experiment was conducted to test the redesigned prototype. The software was altered to reflect the new hybrid layout design while the experimental structure was modified to eliminate flaws detected in the first study. The comments and impressions of a pilot subject contributed to further modifications in the structure of the experiment.

6.1 Changes in Experimental Design

The experimental design for the second study differed from that of the first in a number of ways.

Number of Sessions and Session Length

In an attempt to address the possibility that the first study may not have been long enough to observe an effect[34], the number of sessions was increased from 10 to 15 for the pilot study, and the length of a session was increased from 15 to 20 minutes of text entry. The pilot subject found this duration too long, which led to a change in experimental design from fixed to variable session length (i.e., a fixed number of phrases).

Number of Conditions

The second experiment had only one condition, entering phrases with the redesigned prototype. The results were compared to the fixed QWERTY-condition results of the first experiment.

Monitoring of Subjects

In the first study, all subjects performed the experimental task at the same computer, in the presence of the experimenter. Subjects would arrive to complete sessions that had been scheduled in advance and spaced appropriately in time. In the second study, subjects downloaded the experimental software from a web page, read elaborate instructions, and conducted sessions at their discretion, with the understanding that sessions be spaced by no less than two hours, and by no more than 24.

[34] In fact, the learning model suggests that it would have taken an unrealistically long amount of time to observe an effect, as discussed in Chapter 4. Thus, the length of the first experiment was sufficient, however, at the time this was not known, and untold amounts of cash were needlessly paid to all too willing undergraduates.

Hardware and Operating System Variance

To maintain some consistency of computer hardware and operating system, subjects were required to have the use of a Pentium PC running Microsoft Windows '95 or higher. No users of Macintosh or less powerful PC computers, or with machines running earlier versions of Windows, participated in the study. The software was tested on a number of Pentiums of varying processor speeds and found to visually update instantaneously on all of them.

Remuneration of Subjects

In the first study subjects were paid $8 per session, an amount which took into account the time to perform the experimental task, as well as the inconvenience of having to come to ten appointments. Subjects in the second study, who could perform the experiment in the comfort of their home, were paid a flat sum of $60 for their participation.

Table 6-1: Sample phrases from the first and second experiments

Experiment	Phrases
1	• THE TIME IS ALWAYS RIGHT TO DO WHAT IS RIGHT • PEOPLE JUDGE YOU BY YOUR ACTIONS NOT YOUR INTENTIONS • YOU MAY HAVE A HEART OF GOLD BUT SO DOES A HARD BOILED EGG
2	• IT IS A PLEASURE TO MEET YOU • PLEASE CALL BACK LATER • WHEN YOU WISH UPON A STAR

Stimulus Phrase Length

It had been observed that subjects in the first experiment had trouble remembering entire phrases and thus had to periodically change their focus from the keyboard to the phrase area, and back. Some subjects even fell into a pattern of reading one character at a time, rather than whole words or phrases, thereby reducing the ecological validity of the task. In an effort to eliminate this probable cause of slower entry speeds, the average length of phrase was reduced from 13.9 words in the first study to 5.2 words in the second. The aim was to make the phrases short enough so that subjects would not need to reread them during entry. Table 6-1 shows three phrases from each experiment.

Forced Memorization of a Phrase Before Entry

Entering a phrase from memory more closely mirrors the act of text composition than does reading and entering a phrase in parts. In free composition, the subject composes a phrase in memory and then externalizes it by entering it. Entering a phrase from memory as an experimental task differs only in how the phrase got into memory, through reading as opposed to creation. It was felt that defining the experimental task as entering phrases from memory would improve the ecological validity of the study (Gibson, 1966).

To force subjects to memorize phrases before beginning entry, a number of steps were taken. First, subjects were instructed to memorize the phrase before beginning to enter it. They were encouraged to read the phrase aloud to encode it more deeply, thus facilitating memorization (Craik & Lockhart, 1972). The read-aloud instruction was repeated in a text message at the beginning of every session. Second, rather than keeping the phrase visible while the subject entered it, as had been the case in the first experiment, the phrase would remain on screen until the subject hit the spacebar, indicating readiness to begin. The phrase would reappear in the event of an error--to help the subject resume entry at the correct place--but would disappear again with the next correctly-entered character. As in the first experiment, to avoid strings of consecutive errors, a highlight indicated the next correct letter to enter.

6.2 Pilot Subject Observations

The revised prototype was pilot-tested on a single subject, a 21-year old male undergraduate student in computer science. The subject completed 15 sessions, two of which were interrupted by a bug in the software, which was subsequently fixed. Over the course of his participation, the subject informed the experimenter of several typographical errors in the phrases, and these were subsequently corrected. As mentioned, he found the 20-minute session length too long. He claimed that towards the end of a session he would make a lot of errors, which he attributed to boredom, fatigue and frustration. Since elaborate error data were not recorded for the pilot subject, this observation was impossible to verify.

While reducing the length of sessions was considered, a discussion of other ways to solve this problem ensued. It was noted that while subjects could rest between phrases, they received no feedback on their progress or performance during, or after a session. Many researchers, (e.g., Card, English & Burr, 1978) have found such feedback to be a strong factor in maintaining subjects' motivation.

The addition of such feedback recommended a change to a block design. The interval between blocks provides a natural place for the presentation of feedback. The following design was adopted: 11 blocks per session[35], five phrases per block. Feedback between blocks consisted the number of blocks completed thus far and entry speed on the last block.

Additional Comments

The pilot subject stated that he became totally familiar with the layout corresponding to the beginning of a word after two or three sessions. He did not become familiar with other layouts to the same degree, but did develop a "feel" for letters that occurred most frequently in the fluctuating part of the layout. The subject affirmed the choice of QWERTY for the fixed part of the layout as a good one when he stated that it eliminated any hesitation when moving the cursor outside the fluctuating part. Occasionally, he would begin moving towards a letter in the QWERTY part, then notice that it was in the fluctuating part, and backtrack. In some of these instances, the target was at times not detected at the cursor home position. The reader will recall that this was also observed in the first experiment (see p. 75).

6.3 Method

6.3.1 Subjects

Subjects were solicited by various means, including electronic newsgroup postings, paper notices (Appendix B, p. 118) around the University campus, and word of mouth. Twelve subjects (seven male, five female) were admitted to the study after stating in an email message that they satisfied the following conditions of participation:

- Access to a Pentium computer running Windows '95 or better with at least 8-bit video capability
- Fluency in English
- Strong familiarity with the QWERTY keyboard

[35] The intention was to exclude block 1 from the analysis, leaving ten blocks per session. This eliminates the effect of having to "warm up".

- Availability to perform at least one session on consecutive days until the completion of the last session

Of the twelve who began the experiment, only ten completed it, seven male, and three female.

Several respondents were rejected due to their failure to meet the requirement of access to a Pentium computer. There were also several dual requests to participate by roommates or siblings sharing the same computer. In these cases, only one individual was accepted, to avoid any potential bias.

6.3.2 Apparatus

The hybrid design described on p. 80 was used in the experiment. The software was modified to reflect the new design. In addition, the between-block feedback and post-session performance statistics were incorporated into the software. The redesigned prototype also made use of a new set of alphabetic orderings (see Table 2-5, p. 24) generated from the digram frequencies of the original corpus descirbed in Section 2.2.7 (p. 19).

The self-administered approach did not allow for keyboard preparation as in the first experiment. Thus subjects were given elaborate instructions on a web page, which they were asked to print out. The keys to use for input were described in this document. While the left-handed set were the same as in the first experiment (see p. 55), the right-handed set were changed to the i, j, k, and l, keys for navigation and the spacebar for selection, to avoid excluding people with keyboards lacking a numeric keypad.

6.3.3 Procedure

Respondents who met the conditions for participation were directed to a web page to download the experiment software. They were asked to read the online instructions (see Appendix D, p. 133) carefully before beginning the first session. Key instructions were repeated in a series of windows embedded in the application (see Appendix D, p. 137). There was no practice session or tutorial before the first session, so that initial skill level could be captured as accurately as possible. The instructions stated that subjects should not be concerned if they initially made lots of errors.

There was no fixed condition in this experiment—only the hybrid layout was tested. Each session consisted of 11 blocks of five phrases. The subject would read a phrase, then hit the space bar, causing the phrase to disappear and the keyboard to appear. After five phrases, a message stating the subject's entry speed on that block, and instructing them to rest a while, would appear. After the last block, the program displayed graphs of the subject's performance on the entire session, and on all previous sessions.

Initially subjects were asked to email the data file *after the last session only*. The need to monitor the experiment periodically had not been anticipated, so after some subjects had begun the experiment, the web page of instructions was modified to instruct subjects to email the data file after *each* session.

Subjects completed a demographic questionnaire sent to them by email after the first session (see Appendix D, p. 137), as well as an interview questionnaire sent to them after the last session (see Appendix D, p. 141).

6.4 Results

Table 6-2: Mean entry speed (wpm) by session for hybrid FOCL-QWERTY layout.

Session	*Entry Speed*
1	8.71
2	10.74
3	12.17
4	12.81
5	13.67
6	14.05
7	14.54
8	14.93
9	15.57
10	16.05
11	16.34
12	16.55
13	16.87
14	17.10
15	17.51
16	17.89
17	17.91
18	18.21
19	18.25
20	18.70

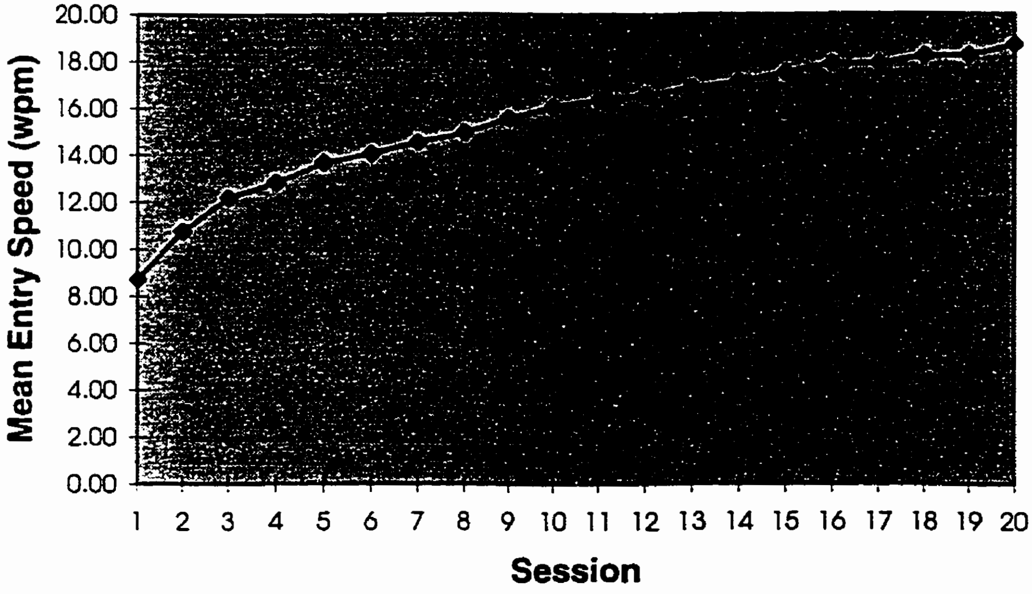


Figure 6-1: Graph of mean entry speed by session

The mean entry speeds for the second experiment are presented in Table 6-2 and displayed graphically in Figure 6-1. The mean entry speed after session 1 of 8.62 wpm is higher than the session 1 mean from experiment 1 (fixed QWERTY condition) of 6.04 wpm. The maximum mean entry speed in experiment 1 of 10.88 wpm is exceeded in the second experiment after just three sessions. The maximum mean entry speed on an individual block was 23.9 wpm. Nearly every subject achieved speeds exceeding 20 wpm on at least one, and typically more than one, block.

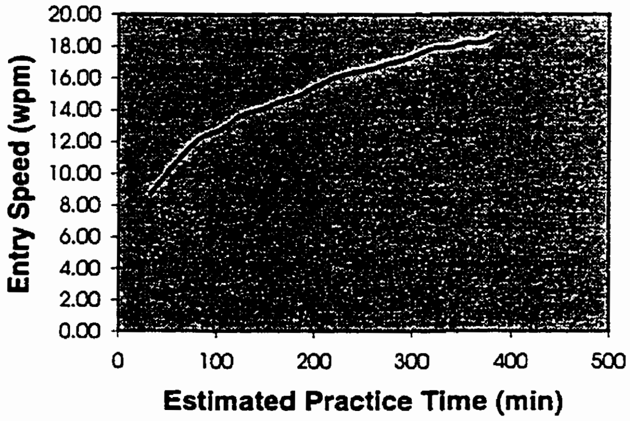
Due to differences in experimental design (see p. 85), the data from the two studies are not directly comparable. In particular, session length in the first experiment was fixed (roughly 15 minutes), whereas in the second experiment it varied. To allow comparison, the data from the second experiment were transformed so that each session mean was associated with an amount of practice time rather than with a session number. This transformation was accomplished by estimating the number of characters entered per session (see equation 6), and using this estimate, and the session entry speed, to calculate the average session length, in minutes (see equation 7).

$$\frac{\#characters}{session} = \frac{\#phrases}{session} \bullet \frac{\#words}{phrase} \bullet \frac{\#characters}{word} = 55 \bullet 5.2 \bullet 5 = 1430 \quad (5)$$

$$sessionLength = \frac{1}{entrySpeed} \bullet 1430 \bullet \frac{1}{60} \quad (6)$$

Practice time is the cumulative sum of session lengths. This provides an x-axis that allows comparison with results from the first experiment. The transformed data, along with a graph, are shown in Table 6-3.

Table 6-3: Mean entry speed (wpm) by estimated practice time (min).



Session	*Practice time*	*Entry Speed*
1	33.17	8.71
2	59.78	10.74
3	83.41	12.17
4	106.17	12.81
5	127.50	13.67
6	148.28	14.05
7	168.43	14.54
8	187.80	14.93
9	206.64	15.57
10	224.87	16.05
11	242.70	16.34
12	260.20	16.55
13	277.40	16.87
14	294.51	17.10
15	311.05	17.51
16	327.55	17.89
17	343.90	17.91
18	359.94	18.21
19	375.94	18.25
20	391.50	18.70

6.4.1 Learning Model

The transformed data were used to determine the learning curve equation.

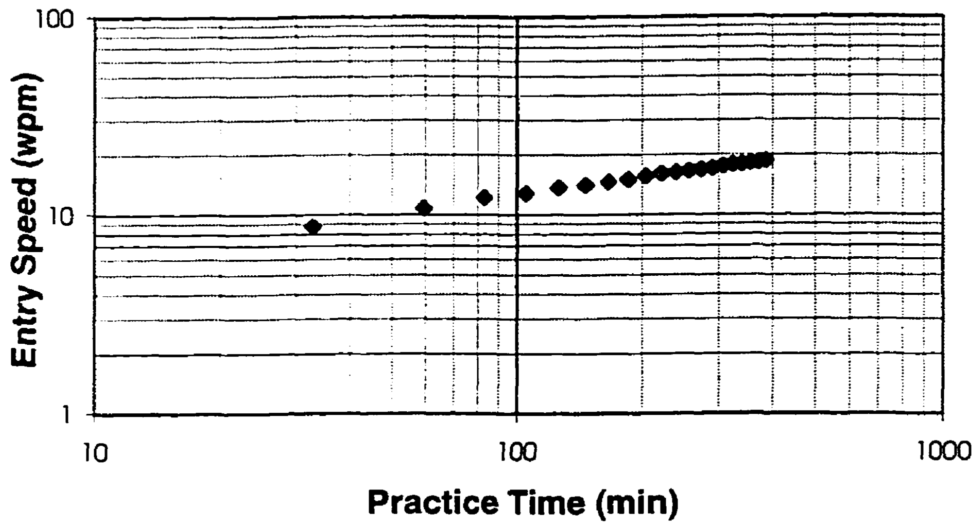


Figure 6-2: Mean entry speed by practice time in log-log coordinates.

Figure 6-2 plots the log-log data of mean entry speed by practice time. The graph illustrates a strong linear relationship, which is confirmed by linear regression analysis ($R^2 = 0.995$). The slope of the regression line (α) is 0.29. Thus, the learning curve for the hybrid prototype can be approximated by the following equation:

$$T_N = 8.71N^{0.29} \tag{7}$$

6.4.2 Error Rate

The mean of mean error rates for each session of experiment 2 is 2.13 %, with a standard deviation of 0.37 %. This is somewhat higher than the equivalent values of 1.55 % and 0.29 % for the FOCL condition in the first experiment, and slightly higher than the mean of 2.05 % observed in the QWERTY condition, though with considerably more variance than the standard deviation of 0.15 % in QWERTY condition error rates. Figure 6-3 graphs mean error rate by session. There is a decrease in error rates in the first few sessions, achieving a minimum at session 5. This is followed by a gradual increase over the remainder of the experiment, with a net improvement of zero as the session 1 error rate is repeated in sessions 16 and 20.

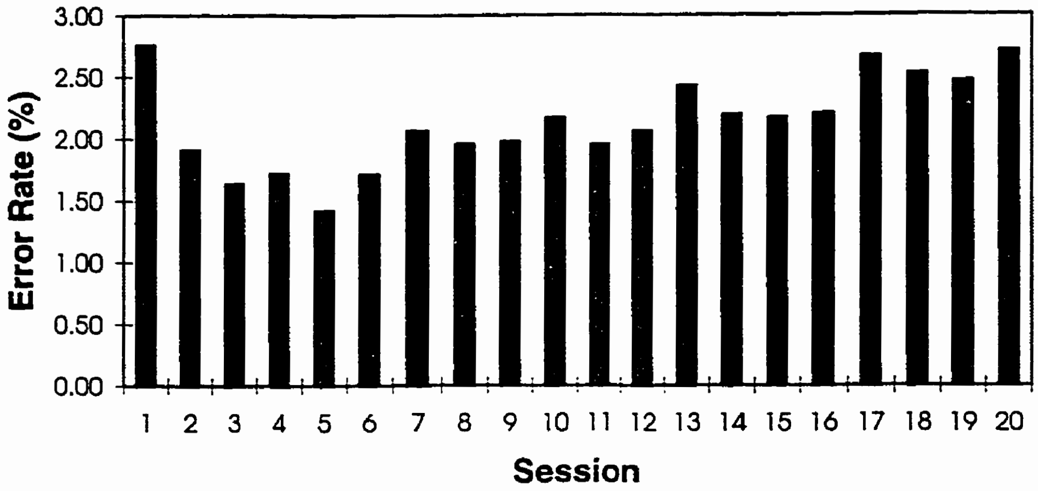


Figure 6-3: Bar graph of mean error rate by session.

6.5 Discussion

Entry Speeds

Transforming the x-axis from session to practice time allows the comparison of mean entry speeds from the two experiments after similar amounts of practice. For example, after 60 minutes of hybrid layout practice (experiment 2, session 2) the mean entry speed was 10.74 wpm, whereas after 60 minutes of fixed QWERTY layout practice (experiment 1, session 4), the mean entry speed was 8.80 wpm, a 22% difference. The mean of 14.04 wpm after 148 minutes of hybrid layout practice (experiment 2, session 6), is 29% more than the mean of 10.88 wpm after 150 minutes (experiment 1, session 10) of fixed QWERTY practice. Figure 6-4 plots the data from the two experiments on a common x-axis of practice time.

The learning model for the fixed QWERTY layout predicts an entry speed of 13.65 wpm after a number of sessions equivalent to the total length of experiment 2. The final hybrid prototype mean entry speed of 18.70 is 37% faster than this predicted value. Thus, in addition to achieving higher entry speeds, the hybrid prototype also achieves better performance at a faster rate, as indicated by increasing differences in means as practice time increases. This accelerated hybrid layout improvement is predicted by the greater slope (0.29 vs. 0.25 words/min^2) of the log-linear learning curve equation of the hybrid layout.

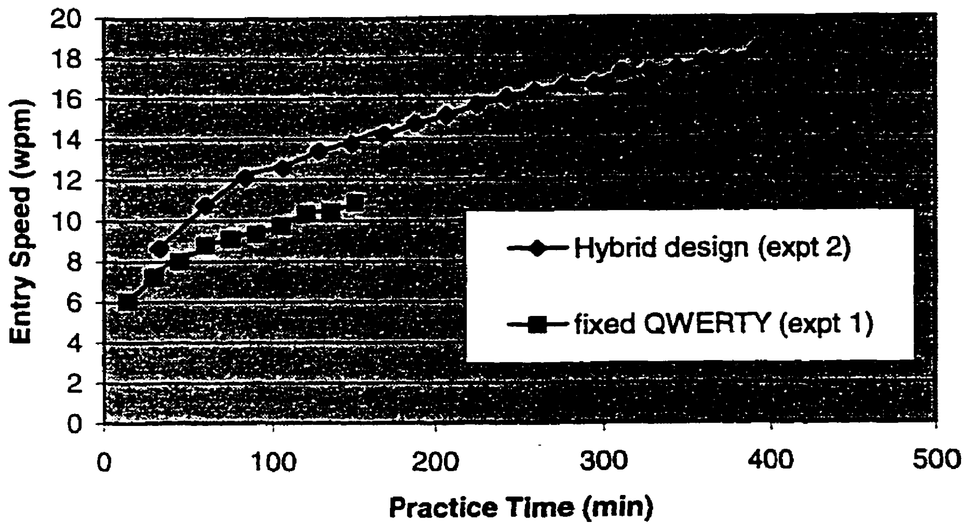


Figure 6-4: Mean entry speed (wpm) for fixed QWERTY condition (experiment 1), and hybrid (FOCL) design (experiment 2), by practice time (min). Hybrid data have been transformed to allow plotting of two data sets along a single axis.

Does this experiment demonstrate the superior performance of the hybrid layout as compared to a fixed QWERTY layout? Due to the many differences in experimental design between the two studies, it is difficult to make a strong claim that the hybrid layout outperforms the fixed QWERTY layout. Better performance in the second experiment may have been due to one or both of the following factors:

- *Subjects were forced to memorize phrases before entering them.* The opportunity to reread phrases at any time during the first experiment may have slowed entry speeds by adding periodic reading time.

- *Subjects received performance feedback between blocks.* Such feedback is known to motivate and have a positive effect on performance (Card, English, & Burr, 1978; Welford, 1968) as subjects try to improve on their last score.

The time stamp data from the first experiment provide an way of assessing the effect of reading time. This is left as a task for further investigation. It is worth noting that the pilot subject performed at a similar level to actual subjects without the benefit of performance feedback.

Reading time might partly explain the lower entry speeds in the first experiment. However, the differences in means are large, and therefore controlling for reading time might not completely eliminate them. Regardless of the outcome of controlling for reading time, the results of this experiment show that FOCL is a viable mobile text entry method. At the very least, it performs no worse than the fixed QWERTY layout. With an array of only nine positions to search *and* the advantage of considerably fewer keystrokes, the hybrid layout is, if not faster for text entry, much less demanding of the user. With mean entry speeds within the range of handwriting achievable after about three hours of use, the technique offers a reasonable learning curve that will not intimidate potential users. The maximum observed entry speed on an individual block of 23.9 wpm, and the many instances of speeds exceeding 20 wpm suggest that FOCL has strong potential. Additional improvements to the design may yield even greater performance improvements.

Error Rates

A possible explanation for the progression of error rates is that as subjects learn the method, their skill improvement leads to fewer errors initially. However, when a certain level of skill is achieved, users become more confident and thus more tolerant of risk, which leads to increased errors. One might expect the hybrid prototype to exhibit error rates between those of the fixed QWERTY and full FOCL layouts, however, the hybrid error rates exceed both. While the hybrid rates remain low, in general, the difference might be explained by the absence of an experimenter, which might lead to less diligence on the part of subjects.

Chapter 7: Conclusions

7.1 Summary

The text entry method described in this thesis is a response to a growing market demand for text-messaging on portable communications devices. Mobile text entry poses a significant design challenge due to the limited output, and particularly input, bandwidth of portable devices. The main goal of this research was to devise a faster method than existing pager and cellular phone text entry methods, and one that at least equals handwriting speeds. After an initial prototype, consisting of a 26-character fluctuating layout, failed to satisfy this goal, a process of redesign was undertaken. The resulting hybrid layout—part fixed, part fluctuating—yielded a considerable performance improvement over its predecessor, thereby achieving the research goal. With mean entry speeds of 18.7 wpm, and maximum speeds of 23.9 wpm, the technique is faster than handwriting. Although further data analysis is necessary to determine whether the hybrid layout is superior in performance (entry speed) to a fixed layout, it is undoubtedly no worse. Moreover, the technique requires far fewer keystrokes, and therefore requires less effort of the user than a fixed layout. By reducing the number of fluctuating characters from 26 to nine, the negative effects of visual search, both on performance and user frustration, have been greatly reduced.

7.2 Limitations of Findings

7.2.1 Experiment Input Style

The input style used in the two experiments--one-handed arrow-key navigation and selection using three fingers and the thumb, may yield very different performance results than if fewer, or different fingers, are used. The button on the device in Figure 1-4 (see p. 9), increasingly common in pagers and cellular phones, is typically operated with the thumb exclusively. Whether or not a thumb-only input style would yield entry speeds comparable to those found in this research, is a topic for further investigation. It is possible to imagine a device that supports an input style similar to that used here. The arrow keys would be on the back of the device, and thus accessible to the fingers, while the selection could remain in front, and thus accessible to the thumb. The mapping of left and right arrow keys would have to be reversed.

7.2.2 Reading Time in the First Experiment

As discussed in the previous chapter, the fact that subjects were able to read and reread phrases while entering them may explain the lower entry speeds in the first experiment. Subjects in the second experiment were forced to memorize phrases before entering them. To resolve this question, a task of further research is to analyze the time stamp data from the first experiment for the presence of long pauses.

7.3 Future Research

7.3.1 Additional Redesign Strategies

The Incremental Approach

One could address the sources of frustration described by subjects (p. 69) on an individual basis. For example, the problem of having to relocate the second letter of a double letter might be solved by having characters repeat if the select key is held down for a certain amount of time. The complaint about not being able to exploit chance adjacencies might be solved by adding a "freeze" feature that momentarily stops the keyboard from fluctuating and does not return the cursor to its home position after a selection.

By addressing weaknesses on an individual basis, however, there is a tradeoff. The user interaction becomes more complex, as users must learn more special cases for entering text. This makes expertise more difficult to achieve, and leads to higher error rates, two strong disincentives to adoption.

It is also wise to ask, before adding a new feature to address a particular source of frustration, whether eliminating it will yield an improvement worthy of the redesign effort. For example, while the double letter problem is annoying to users, double letters do not occur frequently in common English. According to the data of Mayzner and Tresselt (1965) double letters occur with a probability of .019, or less than 2% of the time. As such, it may not be worthwhile to develop a solution to this problem, as it is not likely to affect entry rates in a significant way.

Table 7-1: Visual search literature findings with potential design implications to FOCL

Visual search finding	Design implication or idea
1. It takes longer to locate a target symbol when the surrounding symbols are physically similar to the target. Symbols are more easily located if they look different from their neighbours (Duncan & Humphreys, 1989).	Categorize the letters according to distinct physical attributes (e.g., as below) and ensure that members of the same group are not too close together in a FOCL layout. **round:** Q, O, G, C **diagonal lines:** W, Y, K, X, V, Z, A **exclusively horizontal or vertical:** I, L, T **based on the shape Γ:** P, R, F, E
2. Clutter around a target makes it more difficult to locate (Monk & Brown, 1975)	Leave sufficient space between characters in the display.
3. Search times increase with the amount of irrelevant information displayed (Gordon, 1968).	Avoid unnecessary information on screen, and use sans serif fonts.
4. Given a constant search area, the more dense the concentration of non-targets, the longer it takes, on average, to locate targets (Monk & Brown, 1975).	Allow paging through subsets of the character set rather than displaying it entirely.

Visual search finding	Design implication or idea
5. People tend to fixate around the perimeter of the search area, moving their gaze in a circle around the perimeter (Ford, White & Lichtenstein,).	Prioritize character positioning to take advantage of this.
6. Scanning patterns are influenced by the user's mental map (Chase, 1988).	Given clear instructions on how the FOCL method should be used to best take advantage of its benefits.
7. When colour is part of target specification, search times drop significantly (Chase, 1988).	Assign colours to sets of letters (one colour to a letter would be excessive), so that users can gradually learn to look for the red A, the blue B, etc. The limited resolution and display capability of LCD's on mobile devices make this infeasible at the moment, but perhaps not for long.
8. Letters embedded in words or pronounceable letter sequences are more easily located than letters embedded in random letter sequences, a phenomenon known as the word superiority effect (Massaro, 1973)	Arrange layouts so that adjacent letters form words, or pronounceable letter sequences.

Reducing Visual Search Time

The main disadvantage of FOCL is high visual search times. Another possible redesign strategy is to apply principles gleaned from the literature on visual search to facilitate users locating letters. A study of this literature revealed a number of potentially relevant findings.

Table **7-1** summarizes the findings and describes possible ways of exploiting each one.

Many of the design ideas in

Table **7-1** require a violation of the principle of positioning characters according to probability. As such, there will be a tradeoff in keystrokes. Here, again, we see the keystroke-visual search time conflict at work. An additional caveat is that the findings quoted were drawn from performance under strict, sometimes highly contrived, experimental conditions. As such they must be applied with caution. This redesign strategy would involve numerous design iterations combined with empirical testing and is thus fertile ground for future research.

Higher-order N-gram Data

Higher-order *N*-gram statistics lead to improved predictions. In the case of FOCL, this means fewer excursions into the fixed part of the layout. At present, the fixed part is required for approximately 12% of the characters (see Section 5.1, p. 76). If this could be reduced to as little as 5%, the effect might be significant.

Other researchers (e.g., Darragh, Witten, & James, 1990) have successfully incorporated higher-order *N*-gram statistics into real-time systems. Whether acceptable performance could be achieved on mobile devices is a question for further research. It is probably not necessary to store all possible *N*-grams, since most do not occur (Witten & Bell, 1990). This suggests that memory and processing requirements could be reduced through optimization and clever data-structuring. Another possible approach is to store only the most common *N*-grams, or those with probabilities above a certain threshold. In this case, the system would revert to a lower statistical order in a situation where no higher-order prediction is available.

7.3.2 Alternative Ways of Applying *N*-gram Statistics

Over the course of this research, several alternative ways of applying N-gram probabilities to text entry have either emerged or have been suggested by others. It is worth mentioning the best of these, each of which represents a possible future avenue of inquiry.

Idea #1

The layout is fixed, rather than fluctuating. Instead of moving the cursor around the character set in four directions, there are two buttons. One moves the cursor directly to the *next* character in the probability-sorted alphabetic sequence, the other moves it to the previous. As with the current implementation, a select key enters the currently highlighter letter.

Idea #2

Only a subset of the probability-sorted alphabetic sequence is shown at a time. This requires a key or two for paging through the subsets. If the characters are presented on a single line, then only two arrow keys are needed, otherwise four are required. There is a patent for a cellular phone design that includes such a method (Schroeder, 1995).

Idea #3

This is a variation on #2. The subset of characters are laid out in a shape identical to the configuration of physical buttons on the device, in a one-to-one mapping (Norman, 1988). The user merely presses the button corresponding to the desired character. Thus, there are no movement keystrokes required. However, paging is still required.

7.3.3 Other Input Methods

There is no reason the FOCL technique could not be combined with other input methods than the one discussed here. If snapping the cursor to a home position is eliminated, the technique becomes appropriate for stylus-tapping. By moving probable next characters closer to the last tap, FOCL has the potential to produce better entry speeds, although the problems associated with visual search apply as well.

Navigation of the character set could be accomplished in other ways than through arrow keys. For instance, a two-dimensional pointing device such as a track ball could be used to either move in right-left or up-down directions exclusively, or in any direction. The pointing device could be a combined selection button, thereby saving physical space on the pager or cellular phone. Other input styles may be worthy of investigation.

7.3.4 Other Domains

Telephone Text Entry

As telephones are equipped with increasingly large displays, FOCL might be a good method for accomplishing the frequent text entry tasks required in this domain. The number keypad provides an ideal button setup for FOCL, one that is very similar to the input style used in the two experiments.

Touch-screen Kiosks

FOCL would be an appropriate method to use with touch-screen-based text entry, when a full keyboard is unavailable. For example, in situations where kiosks offer services to consumers, but exposing a keyboard to public use is too prone to malfunction, the condensed button set of FOCL could be an advantage.

Physically-Challenged Users

Word and phrase completion tools make text entry accessible to disabled users. For some users, the constant switching between choosing letters and choosing completions may become tiresome. The simpler interaction of FOCL may make it a desirable alternative to some.

7.4 Final Comments

FOCL is an unusual text-entry method. There is something strange about the character layout changing with each character entered. On the other hand, many have described their experience with FOCL text entry as "fun". The advantages of FOCL are that it is implementable in a small physical space, is easy to learn, is less crude than fixed layout approaches in several recently marketed two-way pagers, and is reasonably fast. Although further research and redesign may ultimately yield a system that is very different from the current instantiation, it is hoped that this thesis has demonstrated the fundamental viability of the FOCL technique.

REFERENCES

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review, 89,* 369-406.

Anstey, P. (1988). How much is enough? A study of user command repertoires. *Proceedings of HCI '88,* 491-508. London: Springer.

Bertram, R. L., Champion, D. F., & Eichorn, L. S. (1998). Mobile client computer with hexagonal keyboard. U.S. Patent Number 5805157.

Boff, K. R., Kaufman, L., & James, P. T. (1986). *Handbook of perception and human performance: Vols. I & II.* NY: Wiley & Sons.

Boff, K. R., & Lincoln, J. E. (1988). *Engineering data compendium: human perception and performance.* Wright-Petterson AFB, Ohio: AAMRL.

Butterbaugh, L. C. (1982). Evaluation of alternative alphanumeric keying logics. *Human Factors, 24(5),* 521-533.

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

Card, S. K., English, W. K., & Burr, B. (1978). Evaluation of mouse, rate-controlled isometric joystick, stepkeys, and text keys for text selection on a CRT. *Ergonomics, 21,* 601-613.

Carroll, J. M., & Rosson, M. B. (1987). Paradox of the active user. In J. M. Carroll (Ed.), *Interfacing thought: Cognitive aspects of human-computer interaction.* Boston: MIT Press.

Chase, W. G. (1986). Visual information processing. In K. R. Boff, L., Kaufman, & P. T. James, (Eds.), *Handbook of perception and human performance, volume II: Cognitive processes and performance.* New York: Wiley & Sons.

Craik, F. I. M., & Lockhart, R. S. (1972). Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behaviour, 11.* 671-684.

Crossman, E. R. F. W. (1959). A theory of the acquisition of speed-skill. *Ergonomics, 2,* 153-166.

Cumming, G. (1984). QWERTY and keyboard reform: The soft keyboard option. *International Journal of Man-Machine Studies, 21,* 445-450.

Darragh, J. D., Witten, I. H., & James, M. L. (1990). The reactive keyboard: A predictive typing aid, *Computer, 23,* 41-49.

De Jong, J. R. (1957). The effects of increasing skill on cycle time and its consequences for time standards. *Ergonomics, 1,* 51-60.

Dewey, G. (1923). *Relative frequency of English speech sounds.* Cambridge, MA: Harvard University Press.

Duncan, J., & Ferguson, D. (1974). Keyboard design and operating posture. *Ergonomics, 17,* 731-744.

Duncan, J., & Humphreys, G. W. (1989). Visual search and stimulus similarity. *Psychological Review, 96,* 433-458.

Dvorak, A. (1943). There is a better typewriter keyboard. *National Business Education Quarterly, 12,* 51-88.

Ford, A., White, C. T., & Lichtenstein, M. (1959). Analysis of eye movements during free search. *Journal of the Optical Society of America, 49,* 287-292.

Friedman, W. F. (1960). *Military Cryptanalysis.* Washington, D.C.: United States Government Printing Office.

Gibson, J. J. (1966). *The Ecological Approach to Visual Perception.* Hillsdale, New Jersey: Lawrence Erlbaum.

Gentner, D. R. (1983). Keystroke timing in transcription typing. In W. E. Cooper (Ed.), *Cognitive aspects of skilled typewriting* (95-120). New York: Springer-Verlag.

Gopher, D., Karis, D., & Koenig, W. (1985). The representation of movement schemas in long-term memory: Lessons from the acquisition of a transcription skill. *Acta Psychologica, 60,* 105-134.

Gopher, D., & Raij, D. (1988). Typing with a two-handed chord keyboard: Will the QWERTY become obsolete? *IEEE Transactions on Systems, Man, and Cybernetics, 18,* 601-609.

Gordon, I. (1968). Interaction between items in visual search. *Journal of Experimental Psychology, 76,* 348-355.

Gould, R. (1988). *Graph Theory.* Menlo Park, CA: Benjamin/Cummings.

Green, D. W., Hammond, E. J., & Supramaniam, S. (1983). Letters and shapes: Developmental changes in search strategies. *British Journal of Psychology, 74* , 11-16.

Hammond, E. J., & Green, D. W. (1982).Detecting targets in letter and non-letter arrays. *Canadian Journal of Psychology, 36*(1), 67-82.

Hick, W. E. (1952). On the rate of gain of information. *Quarterly Journal of Experimental Psychology, 4,* 11-36.

Hyman, R. (1953). Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology, 45,* 188-196.

Ichbiah, J. D. (1995). Method for designing an ergonomic one-finger keyboard and apparatus therefor. *U.S. Patent Number 5487616.*

Keele, S. W. (1986). Motor Control. In K. R. Boff, L. Kaufman, & P. T. James, (Eds.), *Handbook of perception and human performance, volume II: Cognitive processes and performance.* NY: Wiley & Sons.

Kinkead, R. (1975). Typing speed, keying rates and optimal keyboard layouts. *Proceeding of the Human Factors Society's 19th Annual Meeting.* Santa Monica, CA: Human Factors Society.

Krueger, L. E. (1984). The category effect in visual search depends on physical rather than conceptual differences. *Perception and Psychophysics, 35 (6),* 558-564.

Leventhal, L. M., McKeeby, J. W., & Mynatt, B. T. (1991). Screen keyboards: An empirical study of the effects of shape and character layout. *Proceedings of INTERACT '91,* 108-112. New York: North-Holland.

Marmaras, N., & Lyritzis, K. (1993). Design of an alternative keyboard layout for the Greek language. *International Journal of Human-Computer Interaction, 5* , 289-310.

MacKenzie, I. S., Nonnecke, R. B., McQueen, J. C., Riddersma, S., & Meltz, M. (1994). A comparison of three methods of character entry on pen-based computers. *Proceedings of the 1994 Human Factors and Ergonomics Society 38th Annual Meeting,* 330-334. Santa Monica, CA: Human Factors and Ergonomics Society.

MacKenzie, I. S., Zhang, S. X., & Soukoreff, R. W. (1998). Text entry using soft keyboards. [submitted for publication]. (See http://www.uoguelph.ca/~imackenzie)

Massaro, D. W. (1973). Perception of letters, words, and non-words. *Journal of Experimental Psychology,* 349-353.

Matias, E., MacKenzie, I. S., & Buxton, W. (1993). Half-QWERTY: A one-handed keyboard facilitating skill transfer from QWERTY. *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems,* 88-94. New York: ACM.

Maxwell, W. C. (1935). The rhythmic keyboard. *Journal of Business Education, 27,* 327-330.

Mayzner, M. S., & Tresselt, M. E. (1965). Tables of single-letter and digram frequency counts for various word-length and letter-position combinations, *Psychonomic Monograph Supplements, 1,* 13-32.

McCarthy, S. V. (1972). Visual serial search performance for number and letter targets. *Journal of Experimental Psychology, 95,* 233-234.

McCarthy, S. V. (1974). Visual serial search for letters, roman and arabic numbers. *Perceptual and Motor Skills, 38,* 1077-1078.

McCarthy, S. V. (1976). Visual serial search across number and letter sets. *Perceptual and Motor Skills, 43,* 311-314.

McQueen, J. C., MacKenzie, I. S., Nonnecke, R. B., Riddersma, S., & Meltz, M. (1994). A comparison of four methods of numeric entry on pen-based computers. *Proceedings of Graphics Interface '94,* 75-82, Toronto: Canadian Information Processing Society.

McQueen, J. C., MacKenzie, I. S., & Zhang, S. X. (1995). An extended study of numeric entry on pen-based computers. *Proceedings of Graphics Interface '95,* 215-222, Toronto: Canadian Information Processing Society.

Monk, A. (1986). Mode errors: A user-centred analysis and some preventative measures using key-contingent sound. *International Journal of Man-Machine Studies, 24,* 313-327.

Monk, T., & Brown, B. (1975). The effect of target surround density on visual search performance. *Human Factors, 17,* 356-360.

National Highway Traffic Safety Administration (NHTSA). (1997). An Investigation of the Safety Implications of Wireless Communications in Vehicles. *URL: http://www.nhtsa.dot.gov./people/injury/research/wireless/*

Neisser, U. (1964). Visual search. *Scientific American, 210,* 94-102.

Newell, A., & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Eds.), *Cognitive skills and their acquisition.* Hillsdale, New Jersey: Lawrence Erlbaum.

Norman, D. A., (1988). *The psychology of everyday things,* New York: Basic Books.

Norman, D. A., & Fisher, D. (1982). Why alphabetic keyboards are not easy to use: Keyboard layout doesn't much matter. *Human Factors, 24*(5), 509-519.

Posner, M. I. (1978). *Chronometric explorations of mind.* Hillsdale, New Jersey: Lawrence Erlbaum.

Potosnak, K. M. (1988). Keys and Keyboards, In M. Helander (ed.), *Handbook of Human-Computer Interaction,* 475-494, NY: Elsevier.

Pratt, F. (1942). *Secret and urgent: The story of codes and ciphers.* Garden City, New York: Blue Ribbon.

Rabbit, P. M. A. (1964). Ignoring irrelevant information, *British Journal of Psychology, 55,* 403-414.

Redelmeier, D. A., & Tibshirani, R.J. (1997). Association between cellular-telephone calls and motor vehicle collisions. *The New England Journal of Medicine, 336*(7). Waltham, MA: Massachussets Medical Society.

Rochester, N., Bequaert, F. C., & Sharp, E. M. (1978). The chord keyboard. *Computer*, 57-63.

Schroeder, M. K. (1995). User interface for cellular telephone. *U.S. Patent Number 5797098.*

Seibel, R. (1963). Discrimination reaction time for a 1023 alternative task. *Journal of Experimental Psychology, 66,* 215-226.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal, 27,* 398-403.

Shannon, C. E. (1951). Prediction and entropy of printed English, *Bell System Technical Journal, 27,* 50-64.

Shein, G. F., Treviranus, J., Brownlow, N. D., Milner, M., & Parnes, P. (1992). An overview of human-computer interaction techniques for people with physical disabilities. *International Journal of Industrial Ergonomics, 9,* 171-181.

Soukoreff, R. W., & MacKenzie, I. S. (1995). Theoretical upper and lower bounds on typing speed using a stylus and a soft keyboard. *Behaviour and Information Technology, 14,* 370-37.

Underwood, B. J., & Shulz, R. W. (1960). *Meaningfulness and Verbal Learning.* Philadelphia: Lippincott

Wang, Q., Cavanagh, P., & Green, M. (1994). Familiarity and pop-out in visual search. *Perception and Psychophysics, 56 (5),* 233-234.

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

Wilkund, M. E., & Dumas, J. S. (1987). Optimizing a portable terminal keyboard for combined one-handed and two-handed use. *Proceedings of the 31st Annual Meeting of the Human Factors Society,* 585-589.

Witten, I. H., & Bell, T. C. (1990). Source models for natural language text. *International Journal of Man-Machine Studies, 32,* 545-579.

Zhang, S. X. (1998). *A high performance soft keyboard for mobile systems.* Unpublished MSc thesis, University of Guelph, Guelph, Ontario, Canada.

Zelman, J. (1998). Search strategy in alphanumeric arrays. *Studia Psychologica, 21,* 152-154.

Zipf, G. K. (1949). *Human behaviour and the principle of least effort: An introduction to human ecology,* Cambridge, MA: Addison Wesley.

APPENDICES

Appendix A: 27 Possible Layouts in First Prototype

A

N	R	S	I	V	B	U	X	J
T	L	Y	M	G	F	Z	A	O
D	C	K	P	W	H	E	Q	

B

E	U	A	I	B	M	C	G	N
O	L	J	Y	V	D	H	P	X
R	S	T	W	F	K	Q	Z	

C

H	A	E	L	U	S	B	G	P
O	K	R	C	N	D	J	V	X
T	I	Y	Q	F	M	W	Z	

D

E	I	A	U	L	F	W	M	K
O	S	Y	G	T	B	C	P	X
R	D	V	N	J	H	Q	Z	

E

R	N	A	L	M	I	P	O	Q
D	S	T	C	W	X	K	B	Z
E	Y	V	F	G	H	U	J	

F

O	I	E	F	S	B	G	K	Q
R	A	L	Y	C	H	N	V	X
T	U	M	D	J	P	W	Z	

G

E	H	R	U	N	Y	W	F	P
O	I	L	G	B	C	J	Q	X
A	S	T	M	D	K	V	Z	

H

E	A	O	U	W	H	C	G	N
I	T	Y	L	M	D	J	Q	X
R	S	P	B	F	K	V	Z	

I

N	T	L	D	E	K	A	B	I
S	C	M	V	F	Z	U	H	W
R	G	O	P	X	Q	J	Y	

J

U	O	I	C	G	K	N	R	V
E	A	D	H	L	P	S	W	Y
B	F	J	M	Q	T	X	Z	

K

E	I	S	O	M	C	G	K	T
N	L	U	P	D	H	Q	V	X
Y	A	B	F	J	R	W	Z	

L

E	L	O	Y	T	P	V	C	H
I	A	S	F	M	R	B	J	X
D	U	K	W	G	N	Q	Z	

M

E	A	I	S	Y	N	D	J	R
O	U	B	F	T	G	K	V	X
P	M	L	C	H	Q	W	Z	

N

D	G	T	I	C	N	F	V	W
E	O	Y	L	U	H	B	X	P
S	A	K	J	Q	R	M	Z	

O

U	R	M	O	P	D	B	Y	Z
N	T	L	V	I	A	E	J	H
W	S	K	F	C	G	X	Q	

P

E	L	A	U	T	W	B	J	N
O	R	P	H	C	D	K	Q	X
I	S	Y	F	G	M	V	Z	

Q

U	A	C	F	I	L	O	R	V
B	D	G	J	M	P	S	W	Y
E	H	K	N	Q	T	X	Z	

R

E	O	I	Y	U	M	C	F	W
S	A	D	L	G	V	B	J	X
T	N	R	K	H	P	Q	Z	

S

T	E	I	U	K	Y	N	G	J
H	O	S	L	W	B	D	R	X
A	P	C	M	Q	F	V	Z	

T

H	E	A	R	Y	C	N	M	K
O	I	T	W	B	D	Z	P	V
S	U	L	G	F	J	Q	X	

U

T	R	N	C	I	D	Y	Q	J
S	L	M	B	A	O	X	K	V
G	E	P	F	Z	H	U	W	

V

E	I	A	B	F	J	M	Q	T
O	Y	C	G	K	N	R	V	X
U	D	H	L	P	S	W	Z	

W

A	H	E	R	D	U	F	M	V
I	O	S	T	K	G	P	W	Y
N	L	B	C	J	Q	X	Z	

X

T	P	I	U	D	H	L	Q	V
A	C	O	F	J	M	R	W	Y
E	B	G	K	N	S	X	Z	

Y

O	E	I	B	W	M	D	K	R
S	P	L	G	U	F	N	V	Y
A	T	H	C	J	Q	X	Z	

Z

E	Z	A	I	C	G	K	P	T
L	S	O	D	H	M	Q	U	W
Y	B	F	J	N	R	V	X	

SPACE

T	A	S	B	O	D	G	I	V
W	H	M	L	R	Y	K	J	Z
F	C	P	N	E	U	Q	X	

Appendix B: First Experiment Materials

<u>Subject Solicitation Notice</u>

Subjects Wanted for 10-day Computer Experiment

I am looking for subjects to participate in a study comparing 2 methods of text entry for small electronic devices (e.g. pagers).

You would be required to attend 10 sessions of no more than one hour each, on separate, and inasmuch as possible, consecutive days. You would receive $8 per session, for a total of $80, payable upon completion of the last session.

Both non-native and native English speakers are welcome, however strong English proficiency is a requirement for participation.

If you are interested, please contact Tom either by email at bellman@dgp.toronto.edu or by phone at (416) 535-6620. Please feel free to pass this message on to anyone you feel may be interested.

Information Sheet

Thank you for participating in this study, which will form the basis of my masters thesis in human-computer interaction at the University of Toronto, Department of Computer Science. Although I don't want to reveal all the details of the experiment, I hope the following explanation will satisfy your curiosity until the end of the study.

What the study is about
The research is concerned with text entry in devices for which a standard keyboard may be impractical. One alternative is to present the characters on the device's display, and to provide the user with 4 arrow keys to move around them and another key to 'type' the character highlighted by the cursor. In this study, I am comparing, within the approach just described, a fixed layout of the characters to a changing layout. In the changing layout option, the characters are rearranged after each key entered so that those most likely to follow that key are closest to the cursor.

I want to see how you perform with each approach over time. In each session you will spend 15 minutes entering phrases using a fixed layout, and 15 minutes doing the same with a changing layout. You will get a chance to try out both methods before we begin.

What you can expect -- and what I expect
There will be two short interviews during the course of your participation in the study, one after the first session and one after the last session. The purpose of these interviews is to get your impressions of the experience. In addition, I will ask you to complete a personal information questionnaire before we begin today. However, you may refuse to answer any or all of the questions in any of the interviews.

You have my assurance that any information I collect is totally confidential and that your name will never be associated with any specific piece or pieces of information in any publication or talk that may arise from this research.

You are free to withdraw from the experiment at any time, with the understanding that you will not receive remuneration unless you complete all 10 sessions.

As discussed previously, you will be paid $8 per session, with the full payment to be received after your last session.

A few things to keep in mind
1) Like all forms of text entry, the goal is to maximize speed while keeping errors to a minimum. Find a speed at which you make no more than a few errors per phrase. If you are making almost no errors, try to increase your speed. If you are making a lot of errors, you are probably going too fast.

2) Do not be concerned with correcting errors as you enter the phrases. There is no delete function in the software used in the experiment. A sound will inform you when you have made an error, but that is only so that you take notice. You should simply determine what the next letter to enter is and continue on. Try not to let errors bother you.

3) The phrases that will appear for you to enter have no other significance than the fact that they are appropriately long English sentences. You may react to some of them with some emotion or other. I ask that you try to focus on the task of entering the sentences without getting caught up in their meaning.

Tutorial

Walk-through for the Text Entry Experiment

Look at the computer screen. There are 3 areas to take note of:

- the top line of large text, where the phrases to be typed will appear
- the thinner single line field, where what you type will be displayed, and
- the set of letters below it.

Now look at the keyboard. Choose a set of orange keys to use depending on whether you are right- or left-handed. The 4 arrow keys move a cursor around the set of letters. When the cursor is over a letter and you press the key with the orange dot (the select key), that letter is typed (it appears in the thin display).

Try typing the letter 'G'.

The only exception to this is the space character. In the fixed layout condition, the space character occupies the entire 4th row. To type a space, press the down arrow while the cursor is anywhere in the 3rd row. You do not need to press the select key.

Try typing a space.

Now click the button labelled 'dynamic layout'.

In the dynamic layout condition, the space character occupies a column to the left of the letter set. To type a space, press the left arrow while the cursor is anywhere in the leftmost column of letters.

Try typing a space.

Notice that in neither layout condition do you need to press the select key to get a space; just the appropriate arrow key when the cursor is next to the space bar.

Now click the button labelled 'fixed layout'.

Move the cursor around the letter set using the arrow keys.

Now try and type the phrase "THE TIME HAS COME" using the fixed layout. Notice how the next letter to be typed is highlighted on the phrase to guide you.

Click the button labelled 'dynamic layout'.

Now type the same phrase using the dynamic layout. Notice that each time you select a letter the cursor jumps back to the top left corner. This is its home position in the dynamic layout condition.

During each session, with each phrase you finish typing a new phrase will appear on the screen. After 15 minutes the program will pause, while it switches to the other layout condition. At this point you may rest for a moment, or proceed immediately.

Enter text as quickly and as accurately as possible. Try to limit yourself to 2-5 errors per phrase. Adjust your speed so that your error rate is within this range.

When you make an error, the program makes a gentle beep sound.
Do not concern yourself with correcting errors, however. There is no delete function, so in the event of an error simply take note of the hilited letter in the phrase you are typing and proceed by typing that letter.

Demographic Questionnaire

First name: ____________________

Sex: Ô male Ô female

Age (yrs.): ____________________

Is English your first language? Ô yes Ô no

Are you... Ô right handed Ô left handed

Which of the following best describes your level of formal education?

Ô High school
Ô Some post-secondary
Ô College diploma or university degree
Ô Some graduate study
Ô Graduate degree

Main occupation: ____________________

If post-secondary student, what is your field of study? ____________

Are you a touch-typist? Ô yes Ô no

Your estimated typing speed in words-per-minute: ____________

Are you a number keypad typist? Ô yes Ô no

What software, if any, do you presently use on a periodic or regular basis?

Ô Wordprocessing
Ô Spreadsheet
Ô Database
Ô Graphics/Video
Ô Multimedia authoring
Ô Sound/Music
Ô Email
Ô Internet browser
Ô Internet
Ô Games

Have you ever played video or computer games regularly? Ô yes Ô no

If so, do you presently, or was it in the past? Ô past Ô present

Sample of Stimulus Phrases

- THERE IS NOTHING SO UNNATURAL AS THE COMMONPLACE
- HE WHO LAUGHS LAST PROBABLY DOES NOT UNDERSTAND THE JOKE
- WHEN ALL ELSE FAILS READ THE INSTRUCTIONS
- THE CORRECT ADVICE IS TO GIVE THE ADVICE THAT IS DESIRED
- NOTHING IS AS EASY AS IT LOOKS
- EVERYTHING TAKES LONGER THAN YOU THINK
- ANYTHING THAT CAN GO WRONG WILL GO WRONG AND AT THE WORST POSSIBLE TIME
- FRIENDS MAY COME AND GO BUT ENEMIES ACCUMULATE
- SMALL CHANGE CAN OFTEN BE FOUND UNDER SEAT CUSHIONS
- TWO WRONGS DO NOT MAKE A RIGHT
- EVEN THE SMALLEST CANDLE BURNS BRIGHTER IN THE DARK
- WITHOUT FOOLS THERE WOULD BE NO WISDOM
- YOU CANNOT PROPEL YOURSELF FORWARD BY PATTING YOURSELF ON THE BACK
- A CLOSED MIND IS LIKE A TREE WHICH HAS STOPPED GROWING
- IMAGINATION IS MORE IMPORTANT THAN KNOWLEDGE
- THINKING IS THE BRIDGE BETWEEN IGNORANCE AND KNOWLEDGE
- ONLY THE PERSON WHO IS OBLIGED TO EXPAND HIS KNOWLEDGE NEEDS TO THINK
- THERE IS ALWAYS AN EASIER WAY TO DO IT
- EVERYTHING PUT TOGETHER FALLS APART SOONER OR LATER
- THINGS WILL GET WORSE BEFORE THEY GET BETTER
- THOSE WHO FAIL TO PLAN PLAN TO FAIL
- IN ORDER TO GET A LOAN YOU MUST PROVE YOU DO NOT NEED IT
- IN SPITE OF YOUR BEST EFFORTS SOME PLANTS WILL DIE
- THE BEST WAY TO LOSE SOMETHING IS TO STRUGGLE TO KEEP IT
- ONE PLACE WHERE YOU ARE SURE TO FIND THE PERFECT DRIVER IS IN THE BACK SEAT
- THE NICE THING ABOUT TEAMWORK IS THAT YOU ALWAYS HAVE OTHERS ON YOUR SIDE
- BECOMING NUMBER ONE IS EASIER THAN REMAINING NUMBER ONE
- IT IS NOT THAT I AM AFRAID TO DIE I JUST DO NOT WANT TO BE THERE WHEN IT HAPPENS
- ABOUT THE ONLY THING EVER LOST BY POLITENESS IS A SEAT ON A CROWDED BUS
- THE BEST SAFETY DEVICE IN A CAR IS A REAR VIEW MIRROR WITH A POLICEMAN IN IT
- LIFE IS A TRAGEDY FOR THOSE WHO FEEL AND A COMEDY FOR THOSE WHO THINK
- ABILITY IS NOTHING WITHOUT OPPORTUNITY
- ONE OF THE GREATEST JOYS IN LIFE IS DOING WHAT PEOPLE SAY YOU CANNOT DO
- EVERYBODY SETS OUT TO DO SOMETHING AND EVERYBODY DOES SOMETHING BUT NO ONE DOES WHAT HE SETS OUT TO DO
- THE TIME IS ALWAYS RIGHT TO DO WHAT IS RIGHT
- PEOPLE JUDGE YOU BY YOUR ACTIONS NOT YOUR INTENTIONS
- YOU MAY HAVE A HEART OF GOLD BUT SO DOES A HARD BOILED EGG
- AFTER ALL IS SAID AND DONE THERE IS A LOT MORE SAID THAN DONE
- IT IS NOT SIZE OR AGE THAT SEPARATES CHILDREN FROM ADULTS
- MOST PEOPLE DO NOT WANT YOUR ADVICE THEY WANT YOUR SUPPORT
- A GOOD SCARE IS WORTH MORE TO A PERSON THAN GOOD ADVICE
- NO ONE THINKS HE LOOKS AS OLD AS HE IS
- BEING OLD IS NOT BAD IF YOU KEEP AWAY FROM MIRRORS
- THE TRICK IS GROWING UP WITHOUT GROWING OLD

- IT IS BETTER TO BE ALONE THAN IN BAD COMPANY
- AMERICA IS THE LAND WHERE THERE ARE TEN MILLION LAWS TO ENFORCE TEN COMMANDMENTS
- THE GREATEST REMEDY FOR ANGER IS DELAY
- SWALLOWING ANGRY WORDS IS MUCH EASIER THAN HAVING TO EAT THEM
- PEOPLE WITH CLENCHED FISTS CANNOT SHAKE HANDS
- APPETIZERS ARE THE LITTLE THINGS YOU KEEP EATING UNTIL YOU LOSE YOUR APPETITE
- MOST PEOPLE PROVE IT IS NOT NECESSARY TO UNDERSTAND THINGS IN ORDER TO ARGUE ABOUT THEM
- THE WEAKER THE ARGUMENT THE STRONGER THE WORDS
- THE ONLY WAY TO GET THE BEST OF AN ARGUMENT IS TO AVOID IT
- THE PERSON WHO GETS ON A HIGH HORSE IS RIDING FOR A FALL
- OF ALL THE THINGS YOU WEAR YOUR EXPRESSION IS THE MOST IMPORTANT
- THE TWO MOST ENGAGING POWERS OF AN AUTHOR ARE TO MAKE NEW THINGS FAMILIAR AND FAMILIAR THINGS NEW
- AN AUTOBIOGRAPHY USUALLY REVEALS NOTHING BAD ABOUT ITS WRITER EXCEPT HIS MEMORY
- NOTHING DEPRECIATES YOUR AUTOMOBILE FASTER THAN YOUR NEIGHBOUR BUYING A NEW ONE
- PEOPLE WHO FORGET TO TURN OFF THEIR AUTOMOBILE HEADLIGHTS ALWAYS REMEMBER TO LOCK THEIR DOORS
- A TREE NEVER HITS AN AUTOMOBILE EXCEPT IN SELF DEFENSE
- IN MANY CASES THE MOST DANGEROUS PART OF A CAR IS THE NUT THAT HOLDS THE STEERING WHEEL
- SEAT BELTS ARE NOT NEARLY AS CONFINING AS WHEELCHAIRS
- HORSEPOWER WAS MUCH SAFER WHEN ONLY HORSES HAD IT

Post-Experiment Interview

Did you find either of the conditions more or less physically tiring? Or were they equally demanding physically?

Did you experience any physical discomfort during the experiment?

Did you find either of the conditions more or less mentally tiring? Or were they equally demanding mentally?

What was your reaction, if anything, to the content of the phrases, at any time? You can mention specific ones.

How did this affect your ability to perform the typing?

In which of the conditions do you think you typed more words per minute, the fixed, the dynamic or neither?

In which of the conditions do you think you made fewer errors, the fixed, the dynamic or neither?

Do you feel you reached the limit of possible typing speed in the fixed condition, or could you continue to improve with more practice?

If yes, what prevents you from improving anymore?

Do you feel you reached the limit of possible typing speed in the dynamic condition, or could you continue to improve with more practice?

If yes, what prevents you from improving anymore?

Were you relaxed, tense or felt nothing in particular while typing in the fixed condition?

Were you relaxed, tense or felt nothing in particular while typing in the fixed condition?

Which of the two conditions did you enjoy typing in more, and why, or did you find them equally enjoyable?

What improvements, if any, would you make to the dynamic layout approach?

Let's talk about the dynamic condition.

Describe how you would read and type the phrases in the fixed condition, e.g. read a word, type a word; read the whole phrase then type from memory.

If different, describe how you would read and type the phrases in the dynamic condition

Did your strategy in either condition change when I introduced the instruction to read no less than a word at a time?

Describe how you would scan for a letter in the dynamic condition?

Did your strategy change when I explained the best way to scan (from closest to furthest)?

If you used this strategy, how did you find it physically? Mentally?

Were there letters you didn't need to scan for in certain situations because you knew where they were? Which ones in which situations?

Did you find you learned the key patterns for any combinations of two or more characters? Which ones?

Mention any thoughts at all you may have add during the course of the 10 days about the approach, negative , positive or neither.

What problems, if any, do you see with this approach?

Is there anything else at all you would like to mention?

Interview Transcript

Did you find either of the conditions physically tiring?
I found the fixed more tiring, because it involved hitting the same key.

Your fingers got tired?
I think it was more the mind than the fingers. It was just mentally tiring to hit the same key more than once. Whereas I found the dynamic more interesting. It kept me on my toes.

More fun?
Ya, it was more fun. More of a game.

Less predictable?
Ya, it was more of a challenge. There was nothing involved in the fixed one. It was just really tedious.

(Long comment by me about adoption of technology and boredom)
I also found if I made my mistakes in the fixed one, for instance, it was from my mind wandering or just being impatient and trying to rush through.

Do you think that you made more errors in the fixed or in the dynamic?
They were about the same. I think in the...I dunno..it's about the same I think. I have a feeling I made more mistakes in the fixed, though.

Because of your mind wandering?
Ya.

Was there any physical discomfort at all during the experiment.
Eyes.

Why?
Staring at the screen.

Was the text easy enough to read?
Sometimes the 'I' and the 'T'--although I never confused them--they do look similar.

Did you find the letters too close together?
No.

Did you ever find the phrases themselves distracting?
In what way?

What they actually said.
Oh, the message? Not really, no, because they're all sort of the same. Occasionally there would be one that was wittier than the others.

But it didn't affect your ability to enter the text?
Oh no, no, maybe the length. Sometimes I found that it would be a bit daunting initially. Like I remember in the 3rd or 4th or 5th sessions, if it was 3 lines I'd say "uh oh, geez". (laughs), especially in the fixed one.

What would you then do? Would you try and type faster?
At the beginning I think I tried to rush it, but near the end, maybe 6th, 7th session, I just relaxed.

So do you think the variation of length of phrase was maybe too much? It should have been more uniform.
I don't know. I found if I got a long one, then I got a 5 word one, I could zip through it.

Were you trying to go faster and faster? Were you aware that you were typing more phrases from session to session?
Hmm...At the beginning I didn't realize that it was 15 minutes each, I thought it was like 7, 8 or 9. I didn't realize the faster you typed the more you would see. I was trying to type faster, especially in the dynamic, the more challenging. I was trying to improve in the dynamic one more than the fixed one.

Is that because you knew I was investigating that?
Maybe. I was thinking of that. But I mean I was trying to go fast in both.

Do you think you reached your limit in either one?
No. Uh...maybe in the fixed one.

(Exchange about drop in scores on last session)
Actually I was improving in the fixed one too. I was getting used to the space bar gave me trouble at the beginning. So I found I did improve, but I think there's more room for improvement in the dynamic one, 'cause, like today, for instance, I found I could anticipate very underused letters, like Qs and Js.

Which letters and which combinations of letters were you starting to anticipate?
There were a few of them.

Think about all of them.
CH, OUT, AND, YOU, IN.

Now, if you saw those character combinations within a word, like say OUT appears in SHOUT, would you pick up on that?
Oh ya, same with all those. CH, like in children, for instance.

You'd do that quicker?
Automatically.

Like a unit?
Ya, exactly. I'd just press the thumb key.

Did you experience any anxiety during the sessions? Would you describe yourself as relaxed during them?
I don't know. Since it was a game I found myself tense at some points, but not anxious. Just like ready, you know?

How did your read and type the phrases? This is getting at when I introduced that instruction to read no fewer than one word at a time. Before that, let's say, were you reading the whole phrase and then typing by memory, were you reading one character at a time?
It varied. I think only in the first session did I type one letter at a time. Only in the first session, because I was just getting used to the... so it was very plodding.

And you were looking down at your hand at that point?
Ya, because I wasn't familiar with where the fingers should be. And so, it led to mistakes more. I think though throughout I was trying to read words (does he mean a bunch of words?), but sometimes if it was a long one I'd break it up.

If it was a long word?
If it was a long *phrase*. I'd have to break it up.

So you would read what, a few words or...?
Ya, if it was very long, like I'd just start typing without reading the whole thing first. Just do a couple of words at a time.

If you read a couple of words at a time, would you type them all without looking back at the phrase, or would you have to consult?
I'd probably type them all without having to look back.

Did you notice that you would sometimes make an error...?
Ya I did make mistakes.

...if you tried to hold it in memory?
Not so often. But sometimes I'd anticipate words and then make a mistake.

Because you thought the next word in the sentence ought to be...?
Ya, unless I don't know why, for some reason at the very beginning (he means of a session) I'd anticipate. Like yesterday I started typing 'THE' for some reason.

So typically the number of words you would hold in memory was between...?
Trying to think...

What was the least?
I guess one, two, up to five.

What was your scanning strategy in the dynamic condition, when you'd try and locate the next letter? In what direction would your eye move?
It tended to go horizontally more than vertically. (He thinks vertical is optimal?) Sometimes there would be--'t' for example, I might be looking for 't'--for some reason I'd always look right. But then I noticed it so I'd go like this, up and down.

Up and down?
Like closest to the left (cursor).

(I explain optimal scanning)

Are there any improvements you would suggest?
Why is 'I' so far?

Any other thoughts?
About the way the space bar, in the dynamic, like having it on the left, takes getting used to, more than on the bottom.

And how about the transition?
From one condition to the next?

Ya.
Only at the very very beginning (of the experiment).

These are all things I'll consider...
But the other thing though is, in the dynamic, having it on the left makes it more distinct than the fixed one, because you can only go right to get the letters. The left is more distinct.

That's right. You picked up on a design choice I made. In error free usage, you're not going to need the left arrow. If you never have to backtrack...
I did have to backtrack, but ideally..

Occasionally. So it becomes a dedicated space bar. Which was part of my reasoning.
I just thought of that now actually.

What I could have done is positioned the cursor in the bottom left, then I could have put the space bar also below, then again you wouldn't have needed that key.

Oh I see.

That didn't really occur to me at that point. For some reason bottom seemed like a strange place to put the cursor. Maybe it would be better that way to be consistent.

Anything else?
In the fixed one it was very funny having the p--I mean that's how the keyboard is--having it stuck in the corner, because you can't go immediately down, because you have to go left then down. I noticed there were a lot of phrases with the word people, which is very unusual.

That's interesting though, so that if you get stuck in that corner you have to go out then come down to the space bar. I wonder if I could have avoided that too.
Cause p is quite a commonly used letter.

I was just imitating the exact QWERTY layout.
Right.

Appendix C: First Experiment Data

Mean entry speed (wpm) by session for QWERTY and FOCL layout types.

Session	*Layout Type*	
	QWERTY	FOCL
1	6.04	5.31
2	7.22	6.99
3	7.99	7.83
4	8.80	8.47
5	9.10	8.96
6	9.34	8.80
7	9.73	9.35
8	10.33	9.67
9	10.41	10.03
10	10.88	10.36

Mean entry speed (wpm) by sex

	Sex			
	Female		Male	
	Layout Type		*Layout Type*	
Session	QWERTY	FOCL	QWERTY	FOCL
1	5.44	5.01	6.55	5.57
2	6.69	6.81	7.67	7.15
3	7.59	7.55	8.32	8.06
4	8.14	8.13	9.35	8.76
5	8.57	8.91	9.55	9.00
6	8.74	8.77	9.84	8.82
7	9.00	9.04	10.33	9.62
8	9.72	9.37	10.83	9.93
9	9.88	9.91	10.85	10.14
10	10.12	9.98	11.51	10.68

Mean entry speed (wpm) by first language

	First Language			
	English		Other	
	Layout Type		*Layout Type*	
Session	QWERTY	FOCL	QWERTY	FOCL
1	6.11	5.38	5.86	5.12
2	7.36	7.22	6.85	6.39
3	8.06	8.07	7.80	7.21
4	9.02	8.71	8.23	7.84
5	9.26	9.20	8.67	8.33
6	9.46	9.05	9.01	8.12
7	9.92	9.61	9.21	8.67
8	10.58	9.99	9.64	8.84
9	10.77	10.47	9.44	8.87
10	11.25	10.75	9.87	9.33

Mean entry speed (wpm) by discipline type

	Discipline type			
	Science		Arts/Humanities	
	Layout Type		*Layout Type*	
Session	QWERTY	FOCL	QWERTY	FOCL
1	6.79	5.80	5.42	4.91
2	8.03	7.20	6.55	6.82
3	8.85	8.29	7.27	7.45
4	9.48	8.71	8.23	8.27
5	9.88	9.44	8.46	8.56
6	10.04	9.05	8.76	8.59
7	10.40	9.79	9.17	8.99
8	11.15	9.81	9.64	9.56
9	11.14	10.08	9.80	9.99
10	11.51	10.76	10.35	10.03

Mean entry speed (wpm) by typing skill

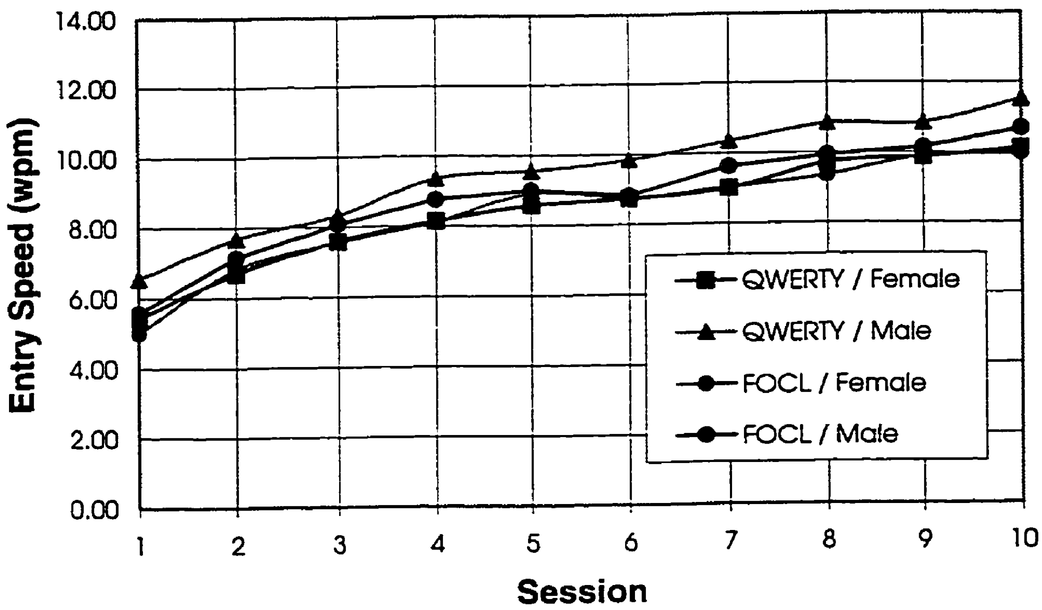
	Typing skill			
	Typist		Non-typist	
	Layout Type		***Layout Type***	
Session	QWERTY	FOCL	QWERTY	FOCL
1	6.40	5.54	4.42	4.30
2	7.69	7.26	5.13	5.80
3	8.40	8.11	6.13	6.58
4	9.13	8.68	7.31	7.55
5	9.48	9.28	7.38	7.54
6	9.75	9.13	7.51	7.29
7	10.03	9.59	8.36	8.29
8	10.77	9.89	8.33	8.71
9	10.78	10.33	8.75	8.72
10	11.22	10.53	9.34	9.59

Mean entry speed (wpm) by video game use

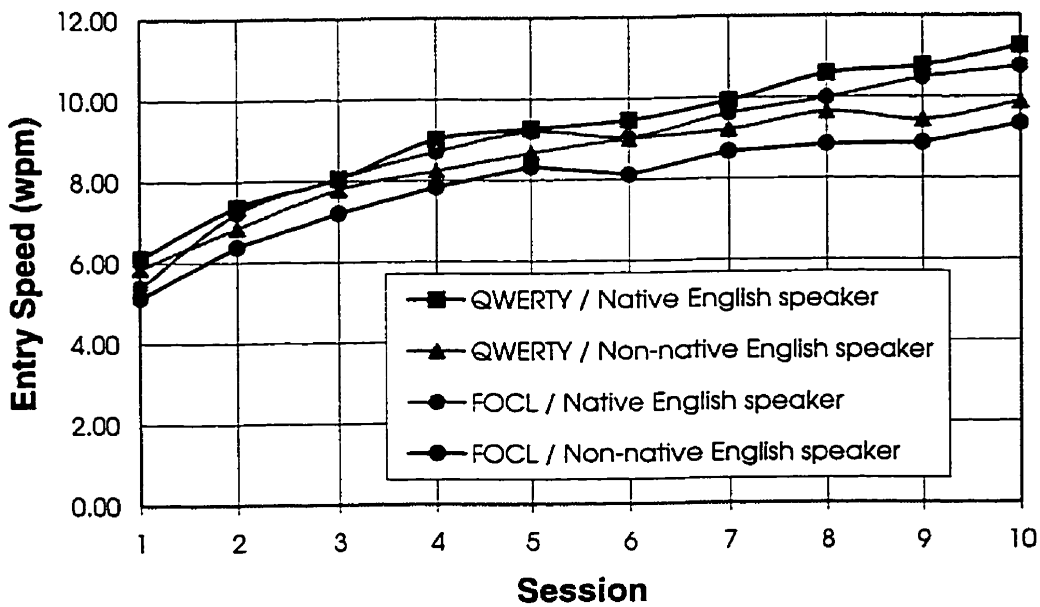
	Video-game use			
	Regular		Never - Irregular	
	Layout Type		***Layout Type***	
Session	QWERTY	FOCL	QWERTY	FOCL
1	6.54	5.66	4.71	4.38
2	7.86	7.29	5.53	6.21
3	8.61	8.29	6.33	6.60
4	9.52	8.98	6.89	7.11
5	9.80	9.45	7.23	7.67
6	10.08	9.36	7.38	7.29
7	10.45	9.91	7.81	7.88
8	11.03	10.18	8.45	8.33
9	11.05	10.58	8.70	8.58
10	11.43	10.91	9.40	8.90

Mean entry speed (wpm) by number-keypad skill

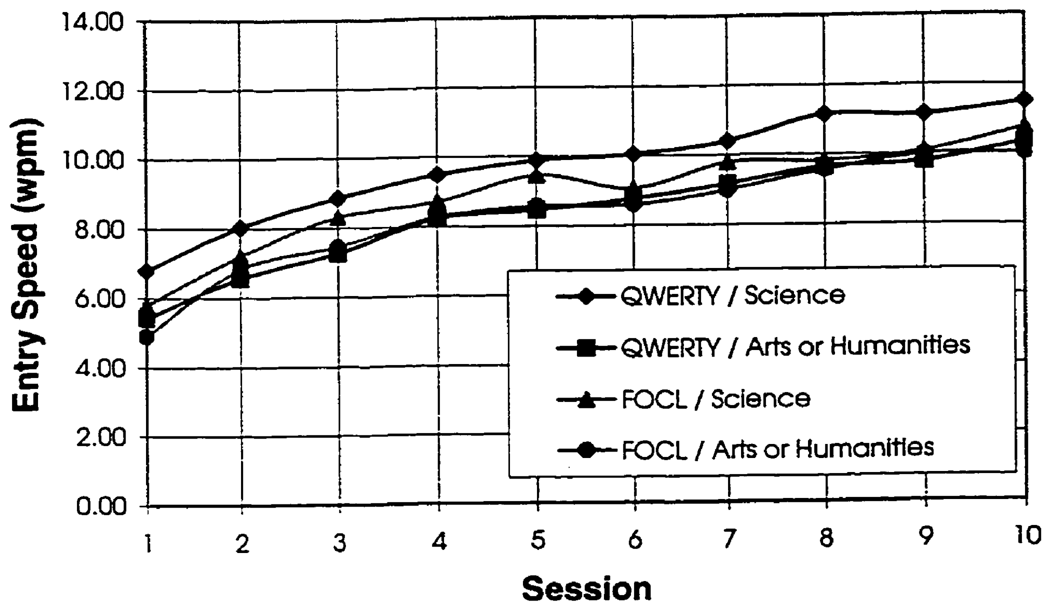
	Number-keypad skill			
	Proficient		Unskilled	
	Layout Type		***Layout Type***	
Session	QWERTY	FOCL	QWERTY	FOCL
1	7.11	6.10	5.44	[illegible].86
2	8.39	7.82	6.56	6.52
3	8.81	8.42	7.51	7.50
4	9.53	8.81	8.39	8.28
5	9.79	9.62	8.71	8.59
6	10.04	9.05	8.94	8.65
7	10.38	9.79	9.36	9.10
8	11.36	9.92	9.74	9.54
9	11.14	10.59	9.99	9.72
10	11.89	10.69	10.29	10.17



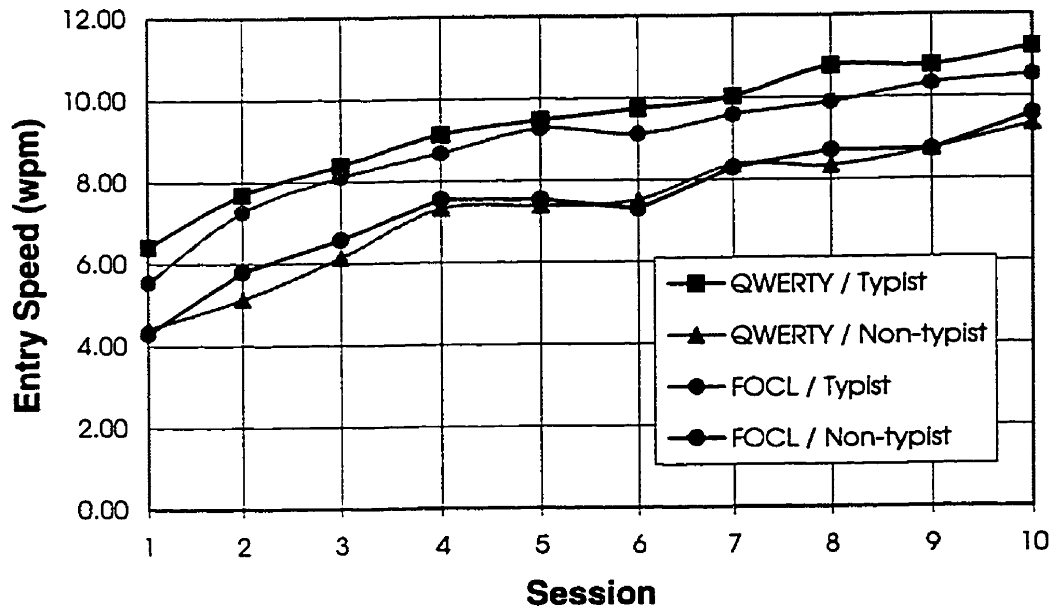
Mean entry speed (wpm) by sex



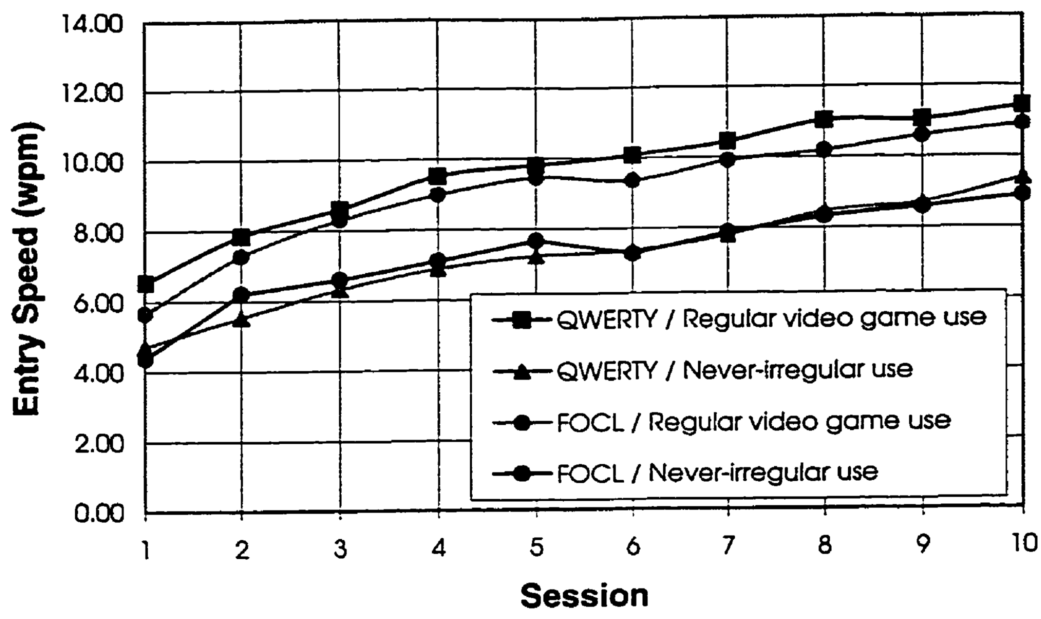
Mean entry speed (wpm) by first language



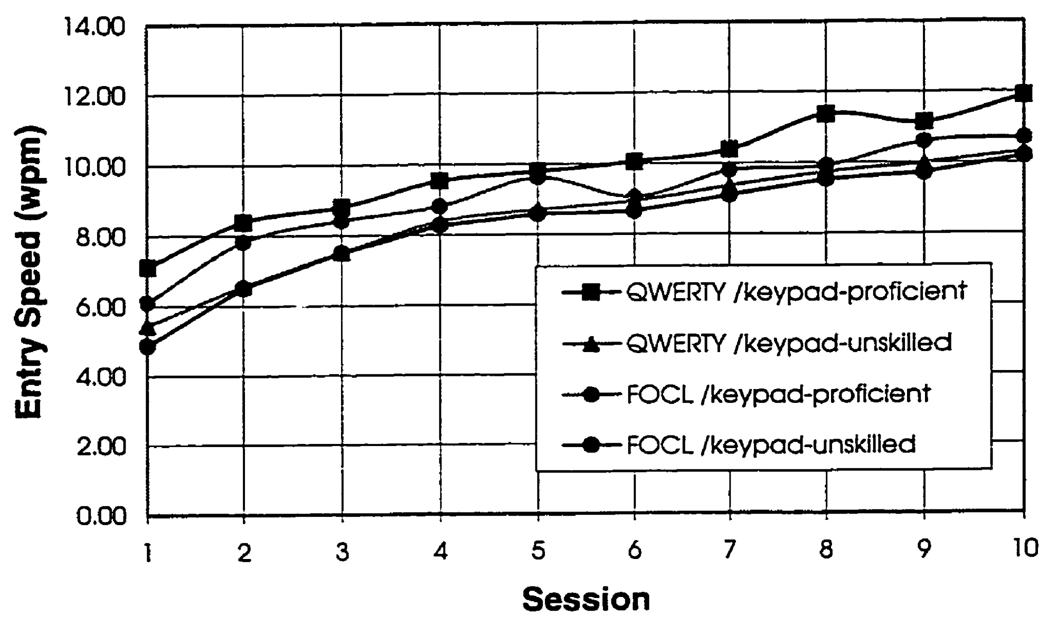
Mean entry speed (wpm) by discipline type



Mean entry speed (wpm) by typing skill



Mean entry speed (wpm) by video-game use



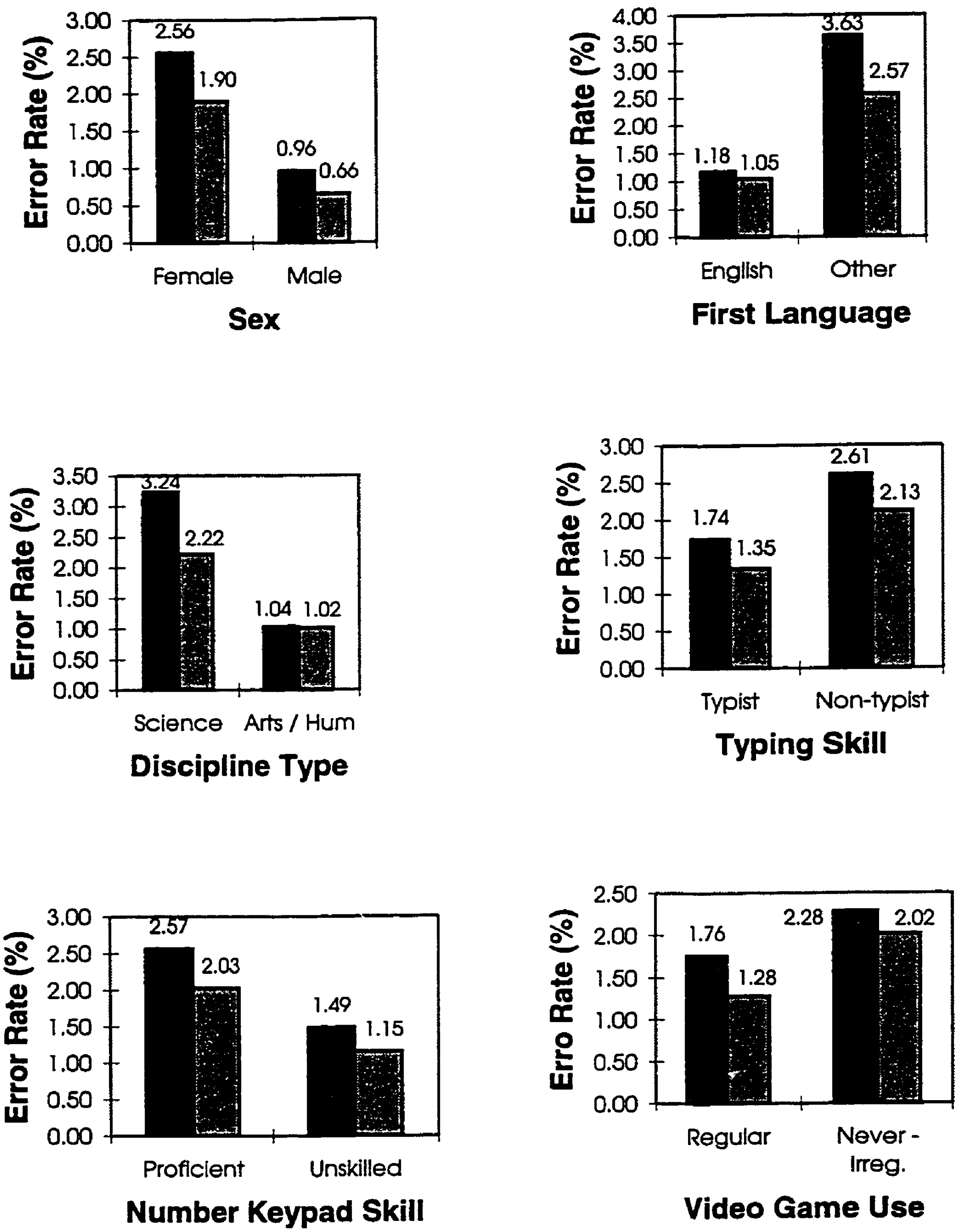
Mean entry speed (wpm) by number-keypad skill

Mean Error Rate (%) by Sex, First Language, and Discipline Type

	Sex		*First language*		*Discipline type*	
Condition	Females	Males	English	Other	Science	Arts/Hum
	(N = 5)	(N = 6)	(N = 8)	(N = 3)	(N = 5)	(N = 6)
QWERTY	2.56	0.96	1.18	3.63	3.24	1.04
FOCL	1.90	0.66	1.05	2.57	2.22	1.02

Mean Error Rate (%) by Typing Skill, Video Game Use, and Number Keypad Skill

	Typing skill		*Video game use*		*Number keypad skill*	
Condition	Typists	Non-typists	Regular	Never-Irreg.	Proficient	Unskilled
	(N = 9)	(N = 2)	(N = 8)	(N = 3)	(N = 4)	(N = 7)
QWERTY	1.74	2.61	1.76	2.28	2.57	1.49
FOCL	1.35	2.13	1.28	2.02	2.03	1.15



Mean error rate (%) by sex, first language, discipline type, typing skill, video game use, and number keypad skill

Appendix D: Second Experiment Materials

Subject Solicitation Notice

Subjects wanted

For an experiment testing a new text entry method

What's required?	Download some software, use it for a while, send in the results
What do you get?	$60.00

Go to **http://www.dgp.toronto.edu/~bellman/pages/exptinfo.html** to see if you meet the few conditions for participation. If you do follow the instructions there to contact me.

Online Instructions

Print this document for future reference

General information

Basically, you will be entering short phrases. You will complete 11 blocks of phrases during each session. There are 5 phrases per block, for a total of 55 phrases per session. The experiment lasts for 20 sessions.

At first each session will take you around 20-30 minutes. By the end you will probably complete a session in around 10 minutes. The software will monitor your speed of text entry, and tell you how you are doing, in words per minute, after each block. A more detailed performance review is presented after each session. You should try to improve your entry speed each time.

After each session, send me an email message with the file FOCLDATA.TXT attached to the message. You will find this file in the same folder as the experiment software. Do not delete this file. If you are unfamiliar with how to attach a document to a message, I can help you if you let me know what software you use to read email. If for some reason you are unable to send the file after a session, e.g., can't connect, don't worry. Just try again after the next session.

I will at some point be collecting basic information from you such as gender and computer experience. You will receive a short questionnaire by email and return it to me completed.

The keyboard

The keyboard is displayed on the screen of your computer (Figure 1) You will notice that it strongly resembles the standard QWERTY keyboard. The main difference is that there are 9 keys inserted between the two sides of the keyboard. Unlike the rest of the keyboard, which is fixed, these keys fluctuate. That is, each time you enter a character, they change. The new characters are those most likely to follow the one you just entered, based on the frequencies of occurrence of all letter pairs, or digrams, in common English.

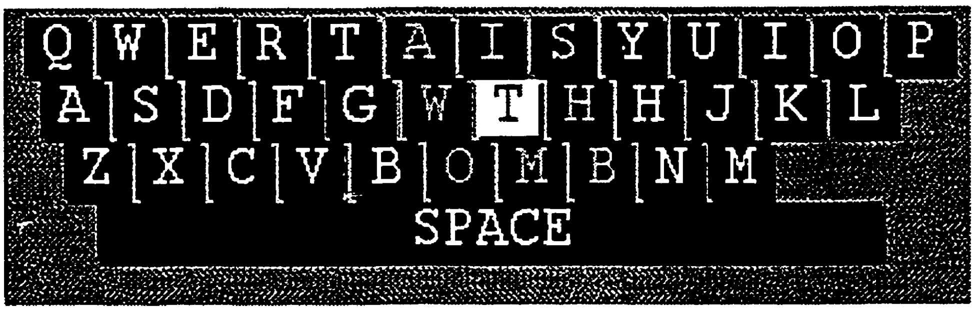


Figure 1: The On-screen Keyboard

To move around the on-screen keyboard you will move a cursor up and down, and left and right by pressing keys on your computer's keyboard. The cursor snaps back to its home position at the centre of the layout--as in Figure 1, where the T is highlighted--after each character you enter.

Left-handed users

If you are left-handed, you will place your index, middle and ring finger on the D, S and A keys respectively (of the physical keyboard). D moves right, S moves down, and A moves left. To move up, hit the W key with your middle finger. So the middle finger covers two keys, W (up) and S (down). Try placing your hand on the keys now tu familiarize yourself with this.

Right-handed users

If you are right-handed, you will place your index, middle and ring finger on the J, K and L keys respectively (of the physical keyboard). J moves left, K moves down, and L moves right. To move up, hit the I key with your middle finger. So the middle finger covers two keys, I (up) and K (down). Try placing your hand on the keys now to familiarize yourself with this.

Selecting letters

You will notice that your thumb falls conveniently on the space bar with your hand in this position. The space bar is for selecting letters that are highlighted by the cursor. When you hit the space bar, the letter at the cursor position appears in an output line containing what you have already entered.

The space character

The space character is treated differently from the other characters. It does not require that you explicitly select it. Rather, just by moving down twice from the cursor home position, the on-screen SPACE key is momentarily highlighted, and then the cursor snaps back to its home position. Remember, two strokes down gets you a space.

Cursor wrapping

If you move all the way to the left or right, the cursor wraps around to the other side of the keyboard. This does not happen if you move all the way to the top or bottom. Moving down, as just mentioned, enters a space character. Trying to move up beyond the first row of the layout has no effect.

Strategy (Very important!)

When entering a phrase, first search for each letter within the 9 fluctuating keys. If it is there, move to it and select it. Only if it is not there, move to where you know it is on the fixed part of the keyboard. You will find that much of the time the letter you seek is in the fluctuating part of the keyboard, and thus no more than 2 keystrokes away.

As you become familiar with the method, you will notice that certain words are very easy to enter. The word 'THE', for instance, is just three presses of the spacebar. Thus, one quickly begins to enter it as such, rather than as three separate letters. Just as in standard typing, try to increase your repertoire of words and letter combinations that you enter as units rather than as individual letters.

Phrase presentation

When you finish entering a phrase, the keyboard is invisible and the next phrase appears above the keyboard area (Figure 2). Read the phrase out loud and memorize it. When you are ready to enter the phrase, hit the space bar. The phrase then disappears and the keyboard reappears. Timing begins with your next key press.

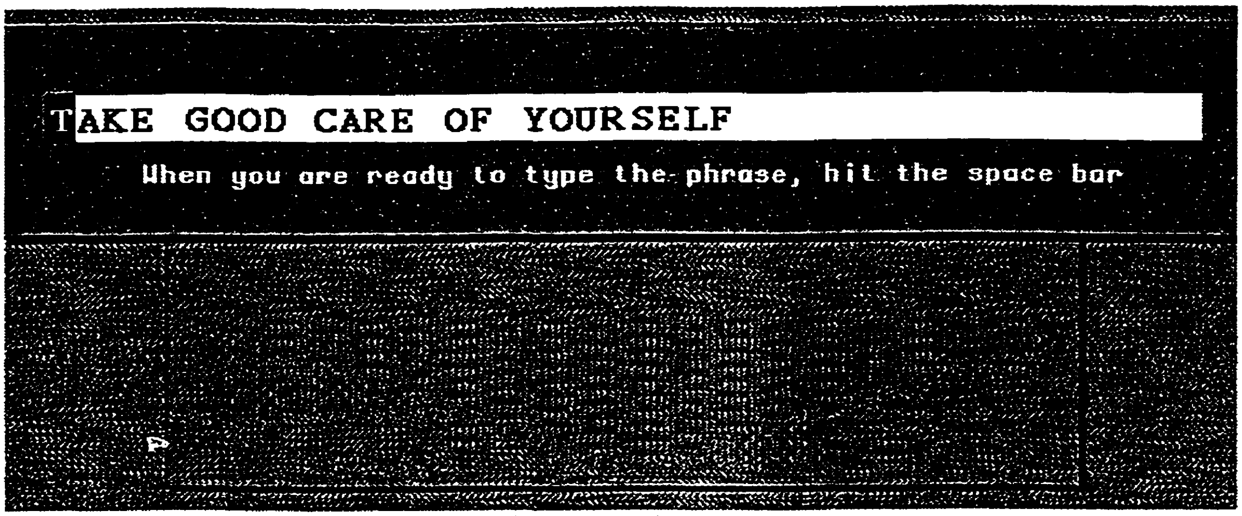


Figure 2: Before you begin entering a phrase

Errors

There is no way to correct errors in this experiment, so don't try to. If you make an error, go right on to the next character. A beep sound will alert you that you have made an error. To make sure you get back on track, the phrase reappears above your output. It will contain a single highlighted character, which is the next character you should enter. For example, in figure 3, the user incorrectly entered N, instead of R, so the phrase reappeared. The highlight indicates that E is the next letter to enter. This approach prevents a string of consecutive errors.

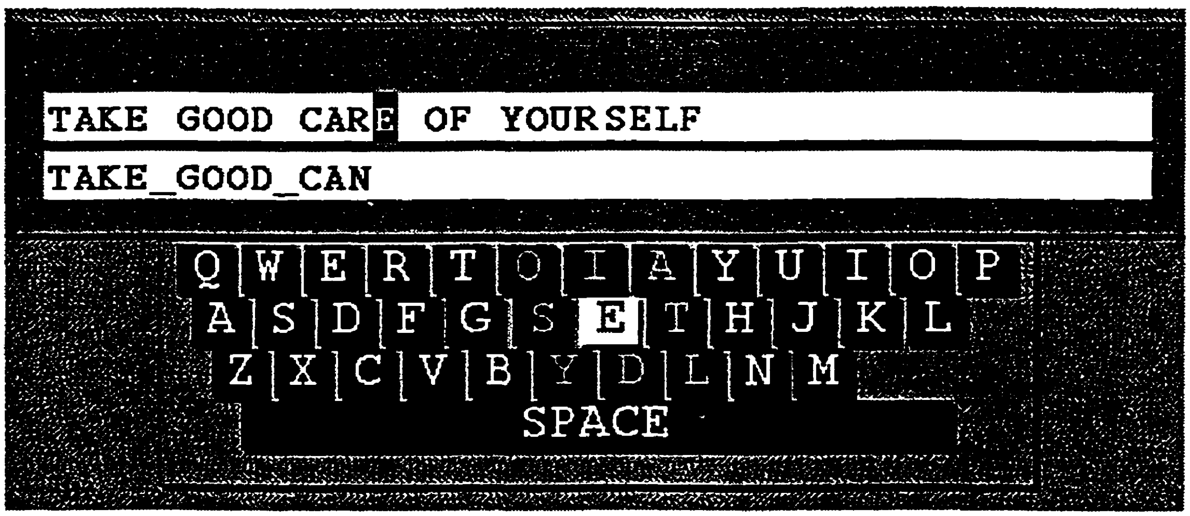


Figure 3: After an error

Resting

Take a break between blocks. You are only evaluated during the time you are actually entering phrases.

Performance statistics

After each block you are told what your entry speed for that block was in words per minute. At the end of a session, before quitting, you can review your performance for that session and for all sessions thus far. This information is presented in bar graph form. You may not always improve from block to block, but your speed will increase consistently over the course of the experiment.

Time between sessions

You should space sessions by no less than two (2) hours and no more than twenty four (24).

Problems

If you encounter any difficulties during the experiment, such as the software crashing, error messages or strange behaviour from the software, please contact me immediately by email (see below) or phone, 416-535-6620.

Finishing

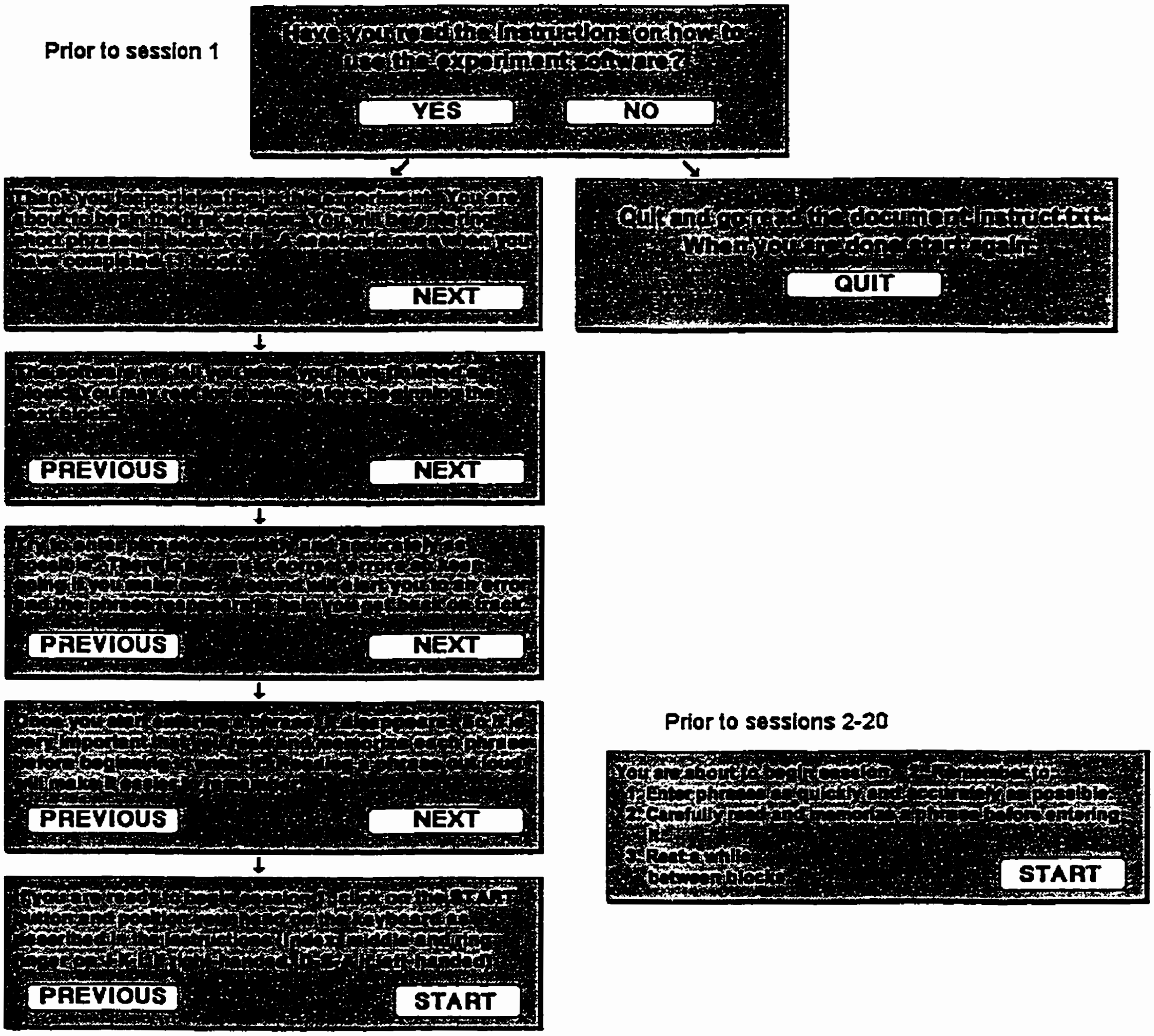
After the last session, as after all sessions, email the file 'focldata.txt' to me at bellman@dgp.toronto.edu. Attach the file to an email message. Please do not delete any files from your machine before you have received confirmation from me that I received your file. I can provide payment of the $60 either by mail or in person.

Before you begin

There is a lot of information to absorb here. If necessary reread this document, or at least skim it, to make sure you understand everything. There is no need to worry, however. Once you start, you will quickly get the hang of it. Don't be concerned if at first you make a lot of errors. In a short time you will be making very few. The initial errors are part of the learning experience I am trying to capture.

To start the first session, double click the file Start.exe in the folder to which you downloaded the experiment software, or, if you created a shortcut, double-click it.

On-screen Instructions Preceding Session 1



Demographic Questionnaire

Please complete the following short questionnaire by including it in an email reply to this message and adding your responses. Although this questionnaire does not collect information of a highly personal nature, I am required by University policy to inform you that you are not obligated to provide any of this information and may leave blank any or all items you deem to be of a private nature. All the information is collected for the purpose of describing the subject sample as a whole, not individual subjects. I am the only person who will view your responses, and will delete your completed questionnaire once I have recorded your responses. I will not link your name to those responses.

Thank you,
Tom Bellman

****

Age (in years):

Sex (M or F):

Your main occupation (if student, please specify your area of study):

City or town you live in (please include province or state):

Are you a touch typist (Y or N)?

Are you right or left handed (R or L)?

Can you perform number keypad entry fast and without looking (Y or N)?

Choose the one statement that best describes you by deleting the other three.
I never play computer or video games.
I very seldom play computer or video games
I occasionally play computer or video games.
I often play computer or video games.

Is English your first language (yes or no)?

List any software you are familiar with and use on a regular basis,
e.g., Microsoft Word, elm, Photoshop, etc.

Estimate the number of hours you spend at a computer on an average day.

How did you find out about this study?

Information about your computer

What is the make, model and processor speed of the computer you are using for the experiment?

What size is the monitor on your computer? e.g., 15 inch.

What version of Windows are you running? e.g., Windows '95.

Sample of Stimulus Phrases

WHAT IS YOUR NAME
HOW OLD ARE YOU
I THINK THEREFORE I AM
WHAT TIME IS IT
ONCE UPON A TIME
ALWAYS LOOK BEFORE YOU LEAP
ABSENCE MAKES THE HEART GROW FONDER
I FEEL MUCH BETTER TODAY
WHAT A BEAUTIFUL DAY IT IS TODAY
I STAYED HOME FROM WORK YESTERDAY
YOUR SHOELACE IS UNTIED
IT HAS BEEN SO LONG
EAT DRINK AND BE MERRY
ALWAYS LOOK ON THE BRIGHT SIDE OF LIFE
FORGET ABOUT THE PAST
YOUR MOTHER IS CALLING YOU
I PREFER TO DINE ALONE
LET ME KNOW WHEN YOU ARE LEAVING
YOU HAVE MY BLESSING
YOU ARE BOTH A GENTLEMAN AND A SCHOLAR
A BIRD IN THE HAND IS WORTH TWO IN THE BUSH
JUST WHAT DO YOU MEAN BY THAT
I COULD WRITE A BOOK
THAT WAS A CLOSE CALL
IT NEVER ENTERED MY MIND
I THOUGHT I SAW A PUSSYCAT
WE HAVE NOTHING TO FEAR BUT FEAR ITSELF
LEAVE THAT MAN ALONE
GET YOUR HANDS OFF ME
THE OLDER I GET THE YOUNGER I FEEL
I LOST IT ALL IN LAS VEGAS
YOU ARE AS YOUNG AS YOU FEEL
AS GOD IS MY WITNESS I AM INNOCENT
YOU HAVE GOT TO BE KIDDING
I HAVE NO ANSWER TO THAT QUESTION
PLEASE ACCEPT MY SINCERE APOLOGY
YOU GO YOUR WAY AND I WILL GO MINE
THE WEATHER TOOK A TURN FOR THE WORSE
I WILL BE AWAY FOR THE WEEKEND
TAKE THE REST OF THE DAY OFF
YOU CAN COUNT ON ME SIR
I HAVE ABSOLUTELY NO REGRETS
YOU MUST STAY FOR DINNER
I SHOULD HAVE KNOWN THIS WOULD HAPPEN
I EXPECTED MORE FROM YOU
THE BEST IS YET TO COME
I HAVE NEVER BEEN IN LOVE BEFORE
THAT IS THE LAST TIME I WILL DO THAT
I FEEL LIKE I AM GOING AROUND IN CIRCLES
HOW DID YOU GET HERE
PUT ME DOWN THIS INSTANT
LONG LIVE THE KING
I FALL IN LOVE TOO EASILY

THAT MAN TRIED TO ROB ME
I TRIED TO WARN YOU BUT YOU WOULD NOT LISTEN
ALL THE WORLD IS A STAGE
IT IS A PLEASURE TO MEET YOU
PLEASE CALL BACK LATER
WHEN YOU WISH UPON A STAR
THINK BEFORE YOU ACT
WAIT FOR ME PLEASE
THERE WILL NEVER BE ANOTHER YOU
WE WILL NEVER SURRENDER
FIGHT FOR WHAT YOU BELIEVE IN
A MIND IS A TERRIBLE THING TO WASTE
SHE RESCUED ME FROM THE COLD
WERE IT NOT FOR HIM I WOULD NOW BE DEAD
ALL MEN AND WOMEN ARE CREATED EQUAL
THAT CHILD HAS THE SWEETEST SMILE
WHEN WOULD YOU PREFER TO MEET
I KNOW JUST THE PLACE
HE HAS A FLAIR FOR THE DRAMATIC
YOUR FLY IS OPEN
YOUR PANTS ARE FALLING DOWN
THERE IS NO TIME LIKE THE PRESENT
I HOPE YOU ARE SATISFIED
HOW CAN YOU LIVE WITH YOURSELF
I NEVER MAKE THE SAME MISTAKE TWICE
I LIKE TO DRIVE WITH THE RADIO ON
I HATE TO SAY GOODBYE
I REFUSE TO ANSWER THAT QUESTION
HAPPY DAYS ARE HERE AGAIN
YOUR SECRET IS SAFE WITH ME
OUR CONVERSATION MUST NEVER LEAVE THIS ROOM
THEY SPEND EVERY MINUTE TOGETHER
WHAT HAVE YOU GOT TO LOSE
THE END JUSTIFIES THE MEANS
MARY HAD A LITTLE LAMB

Post-Experiment Questionnaire

I would greatly appreciate your feedback on the experiment and the text entry method. Please add your responses to the following few questions in an email reply.

*****

1. How would you describe the overall experience of entering text with this method? Was it enjoyable, frustrating, fun, difficult, etc.? Please elaborate if you wish.

2. Was there anything that frustrated you about the text entry method?

3. List any words or letter combinations you had learned to enter as a unit rather than as individual letters (e.g. THE) by the end of the experiment. (as many as you can recall)

4. Did you find you made more or fewer errors at certain times, and were you aware of any factors influencing this?

5. Did the statistics presented between blocks and at the end of sessions motivate you in any way?

6. Did the between-block statistics give you a sense of how much time remained in the session?

7. Do you have any suggestions for improving the design?

8. Would you use such a method of text entry to compose messages if it were available in a device such as a pager, or cellular phone? If no, why not?

Thanks again for your participation.

Appendix F: Output from Chunk Detection Program

$t = 0.2$ s	Session					
Subject	5	6	7	8	9	10
1	Not Available	Not Available	None	None	None	None
2	None	None	None	None	None	None
3	None	None	ETHE 1 THE 4	None	GTHE 1 THE 3	HE 2 THE 4
4	TH 2	HE 2			HE 3	HE 2 THE 1
5	None	None	None	None	None	None
6	OU 2 STHE 1 TH 3 THE 5 THERE 1	THE 3 THERE 1	TH 3 THE 5 YOU 1	OU 7 TH 2 THE 10 THERE 1	ND 2 OU 3 THE 3	THE 8
7	None	None	HE 3 OT 2	RTHE 1 THE 2	HE 3 THE 5	HE 2 THE 9
8	None	None	None	None	None	None
9	ETHE 1 HE 2 OU 2 TH 2 THE 5	OU 4 STH 1 TH 5 THE 3 YO 2 YOU 1	OU 6 RTHE 1 TH 7 THE 1 YOU 1	HE 4 TH 9 THE 10	ER 5 ETHE 1 OU 5 TH 6 THE 7 YOU 1	ER 3 HE 3 ND 2 OU 2 TH 7 THE 5 YOU 1
10	None	None	None	None	None	None
11	HE 3 OU 2 THE 4	HE 4 OTHE 1 OU 3 THE 2 YO 2 YOU 2	THE 4	ARF 1 THE 5 YOU 1	ETHE 1 HE 3 ITHE 1 OU 4 THE 5	OU 5 THE 19 YOU 1

$t = 0.3$ s	Session					
Subject	5	6	7	8	9	10
1	Not Available	Not Available	None	None	HE 3	None
2	None	None	None	None	None	None
3	Not Available	EO 2 ND 2 OU 3 THE 2	ETHE 1 OU 3 TH 2 THE 5 OK	None	OU 2 THE 7	THE 6
4	AND 1 HE 2 IN 2 OU 5 TH 10 THE 7 YOU 1	AND 2 ER 2 ETHE 2 GTHE 1 TH 7 THE 5 YOU 1	AND 2 CH 2 ER 2 KTHE 1 OU 3 RTHE 1 TH 4 THE 6 YOU 2	CH 2 OU 4 OUT 2 TH 3 THE 9 TO 3 YOU 2	AND 1 ATHE 1 ER 4 GTHE 1 LTHE 1 NG 2 OU 2 OUT 1 TH 10 THE 7 YOU 2	AND 2 ER 4 ING 3 OU 4 TH 3 THE 12 TO 2 YOU 5
5	RI 2	None	THE 1	None	THE 1	THE 1
6	IS 2 OU 2 STHE 1 TH 2 THA 1 THE 5 THERE 1	IS 2 NG 2 OU 2 THE 4 THERE 1 YOU 1	HA 2 IS 2 TH 2 THE 7 TO 2 YOU 2	NP 2 OU 8 TH 3 THE 10 THERE 1 YOU 2	ND 2 OU 2 TH 2 THE 3 YOU 2	HA 2 OU 2 THE 10
7	None	HE 4	HE 4 OTHE 1	HE 4 THE 6	BTHE 1 THE 6	HE 2 THE 10
8	ET 2	None	None	HE 2	HE 5	HE 4
9	ER 3 RE 2 TH 3 THE 8 YOU 4	LTHE 1 OU 2 TH 3 THE 7 YOU 5	OU 3 RTHE 1 TH 3 THE 6 YOU 7	AND 1 ER 2 HEB 1 TH 4 THE 18 YOU 1	ER 5 PER 1 TH 9 THE 11 VER 1 YOU 5	AND 3 ER 8 TH 6 THE 11 YOU 3
10	Not Available	None	None	ETH 1	None	TH 2
11	NTHE 1 OTHE 1 OU 2 THE 4 YOU 2	HE 4 OTHE 1 THE 5 YO 2 YOU 5	THE 4 YOU 5	ARF 1 HA 3 HE 3 THE 5 YOU 2	AYOU 1 CTHE 1 TH 2 THE 8 TO 3 YOU 3	AND 1 HTHE 1 THE 19 YOU 5

$t = 0.4s$	Session					
Subject	**5**	**6**	**7**	**8**	**9**	**10**
1	Not available	Not available	HE 2	HE 5 THE 1"	HE 9	HE 3 THE 1
2	None	HE 2	None	HE 2 SU 2	AS 2 EA 3 HA 3 HE 2 NG 3 RO 2 SI 2	HA 2 HE 4
3	Not available	ND 2 NG 3 OU 2 TH 2 THE 2 YOU 1	ETHE 1 OU 3 TH 2 THE 5	ER 3 NG 3	HA 2 OU 3 TH 2 THE 7 TO 4	AR 2 NG 3 THE 6 TO 2
4	AND 2 IN 2 OU 4 OUT 1 STHE 1 TH 13 THE 8 TTHE 1 YOU 2	AND 2 CH 2 ER 4 ETHE 1 HER 1 NG 2 TH 8 THE 7 TO 2	AND 3 CH 3 ER 5 ERE 1 GTHE 1 ITHE 1 OUR 1 RE 2 TH 8 THE 7 TO 2 YOU 3	CH 3 HOUT 1 NG 2 OUT 1 TH 8 THE 10 TO 5 YOU 5	AND 2 ATHE 1 DO 2 ER 5 ERE 1 IS 3 LTHE 1 NG 5 OUT 1 STHE 1 TH 12 THE 8 TO 4 YOU 3	AND 2 AON 1 AR 2 DTHE 1 ER 5 ERE 1 GTHE 1 IN 5 ING 4 IS 3 LY 2 OTH 1 OU 4 TH 3 THE 11 TO 3 YOU 3 YOUR 3
5	HE 3 OU 4 THE 1 YOU 1	OU 4 THE 5	THE 3	THE 4 YOU 1	OU 2 TH 2 THE 5 YOU 4	HE 2 THE 3
6	AND 2 CA 2 IS 6 LO 3 MAK 1 OR 4 ORE 1 PL 2 TH 2 THA 3 THE 6 THEREW 1 VI 2	AND 1 FI 2 HA 4 IS 3 NG 3 OF 3 OU 2 RO 2 SO 2 TE 2 THA 1 THE 4 THERE 1 TO 2	FI 3 HA 4 HAR 1 IS 4 MA 3 NG 3 SIS 1 STHE 1 THA 1 THE 5 TIS 1 TO 5 VTHE 1 YOU 4	EN 2 HA 2 IS 3 NG 2 OF 2 OU 3 TH 2 THA 2 THE 11 THERE 1 TO 3 YOU 8	AND 1 AR 2 CA 3 DTHE 1 IND 1 IS 3 KTHA 1 LO 2 MA 3 NG 2 OF 2 OM 2 OU 3 TH 2	CA 3 DAR 1 EAT 1 EO 2 GHT 1 GTH 1 HA 2 ILL 1 ITHE 1 NG 3 OFTHE 1 RI 2 RO 3 RTHE 1

$t = 0.4$s	Session					
Subject	**5**	**6**	**7**	**8**	**9**	**10**
	YOU 3	YOU 2			THE 2 TI 2 TO 2 TOL 1 TOO 1 TYOU 1 YOU 2	SYOU 1 TEA 1 TH 2 THA 1 THAT 1 THE 8 UR 2 WH 2 WHA 1 YOU 2
7	AND 1 AT 2 EN 3 HE 10 NG 2 NO 2 OR 3 RE 2 RO 2 THE 3	HA 3 HE 4 MA 2 NG 2 TE 2 THE 1 WHIL 1	CA 2 HA 3 HE 4 HO 2 MA 3 RI 4 RO 2 THE 1 WHA 1	AR 2 CA 4 EN 3 FO 2 HE 2 HI 2 IT 2 OR 2 SI 2 SO 2 THE 8 TLY 1 WH 2	AR 2 AS 2 BTHE 1 EA 2 EN 2 HA 2 HEV 1 IS 2 KISA 1 MA 2 NO 2 RO 3 TE 2 THE 6 TMO 1 WH 2 WOR 1	DPA 1 EN 3 GE 2 HA 2 IS 2 IT 3 OR 2 THE 10 TI 2 UTHE 1 WH 7
8	RO 2	CA 2 HE 3 NG 2 RI 3	CA 2 HA 2 KI 2 MA 2 SI 2	AR 2 DCO 1 DID 1 EN 2 HA 2 HE 5 NG 3 NT 2 TE 2 WH 3	AL 2 HE 6 HEN 1 MI 2 NG 3 OR 2 OU 3 RI 2 SSI 1	EN 6 HA 4 HE 2 IT 3 NG 4 NTHE 1 RO 4 TI 3 WH 3
9	ER 3 HA 2 ITHE 1 NTHE 1 RE 2 TE 2 TH 3 THE 6 YOU 5	BER 1 HA 2 TER 1 TH 3 THE 7 THEO 1 YOU 6	NG 4 OU 2 RI 2 TH 3 THE 7 YOU 7 YOUA 1	AND 1 ARD 1 ER 2 FTHE 1 OL 2 OTHE 1 RYOU 1 TH 3 THE 15 THES 1 YOUR 1 YTH 1	CA 3 EN 2 ER 4 ERI 1 IN 2 ITH 1 LO 2 LWH 1 NTHE 1 PER 2 RI 2 STHE 1 TH 7 THA 1 THE 8 THETHIN	AND 3 DHA 1 ER 8 FYOUR 1 GE 2 HA 3 IN 2 ORE 2 RIS 1 STHEMA 1 TH 5 THA 1 THE 10 TO 2 YOU 2

$t = 0.4s$	Session					
Subject	**5**	**6**	**7**	**8**	**9**	**10**
					1 VER 1 YOU 5	
10	Not available	OU 2 TH 3	OU 4 TH 5	ETH 1 OTHE 1 TH 5 THE 4	OU 2 THE 2	HA 2 LTH 1 NG 2 OU 3 TH 9 THE 4
11	DIT 1 HA 2 HE 2 HO 2 OR 2 OU 2 THE 6 THEO 1 YOU 2	HE 3 THA 1 THE 8 YOU 6	HA 2 HE 2 NG 2 TH 3 THE 4 YOU 5	HA 2 HE 4 ND 2 THA 3 THE 8 YOU 3	EISA 1 HE 2 IS 4 IT 2 TH 2 THE 9 TO 4 YOU 5	AND 1 IS 2 THA 1 THE 17 THEE 1 THEM 1 THEMO 1 TO 4 YOU 6 YOUH 1 YSA 1