Modeling of User Performance with Computer Access and Alternative Communication Systems for Handicapped People

Heidi M. Horstmann, M.S. and Simon P. Levine, Ph.D.

Rehabilitation Engineering Program
Department of Physical Medicine and Rehabilitation
University of Michigan

ABSTRACT

Disabled individuals who cannot use a standard keyboard require a special interface in order to use a computer. The GOMS model is used here to quantitatively evaluate three interfaces currently used in computer access systems for handicapped people. Each interface uses a row/column scanning technique for letter selection, and two of the interfaces employ word prediction in an attempt to improve text input rate. Techniques for modeling these interfaces are presented, and the resulting predictions for performance time, learning time, and working memory requirements are discussed. The models predict that the systems with word prediction actually have lower performance than one that allows only single letter selections. Factors contributing to this result include additional mental operators required for use of the word predictive interfaces and an insufficient probability of successful word prediction.

INTRODUCTION

The personal computer has tremendous potential for improving the functional abilities of physically and cognitively disabled individuals. Some of this potential has already been realized, and many new educational, vocational, and recreational opportunities have opened up for disabled individuals through the use of the computer. For a computer to be useful to disabled individuals, alternatives to the computer's hardware or software must often be developed. For example, a disabled user who cannot physically use the standard keyboard must have an alternative means of accessing the computer, referred to as a computer access system. In addition, use of the computer as an alternative communication aid for people who cannot speak requires a special user interface design, similar to that of a computer access system.

This paper addresses issues surrounding the design of these user interface alternatives. The GOMS (Goals, Operators, Methods, Selection Rules) model (Card, Moran, & Newell, 1983) is used to quantitatively describe and predict user performance for three interfaces currently used in computer access and alternative communication systems for handicapped individuals.

BACKGROUND

Handicapped individuals with physical impairments may need an alternative to the standard keyboard for computer access. The exact nature of the physical impairment determines what type of physical input technique is used (e.g., single switch, expanded keyboard). This in turn determines the physical component of the user's "typing" rate. Some users may have cognitive and/or perceptual impairments as well, which affect the mental component of their performance.

There are many communication and computer access aids that are either commercially available or in the final stages of testing, with more packages being developed each year. These incorporate a wide range of physical input methods, such as expanded keyboards, head pointing devices, and breath-controlled switches. In addition, a variety of methods designed to enhance rate, such as symbolic encoding, abbreviation expansion, and word prediction can be used. Unfortunately, developers' publications give only minimal attention to an analysis of their design goals and design decisions. Analyses of these issues that do exist focus almost exclusively on physical efficiency, without considering the mental load on the user in a rigorous or quantitative way (Goodenough-Trepagnier et al., 1982; Rowley, 1987).

METHODS

The GOMS Model

The GOMS model was developed by Card, Moran, and Newell (1983), and refined by Polson and Kieras (1985), among others (1986). The user's behavior is represented by a sequence of elementary steps (called "Operators") defined by the goals of the user and the constraints of the task. The final model is a list of statements that represent the Goals, Methods, Operators, and Selection Rules to provide a complete model of the user's behavior in pursuit of the overall goal, specifying each required step in the proper sequence.

The GOMS model can be used to predict both learning and performance times, as well as points of excessive long or short term memory load. These predictions can then be used during the design process to estimate the consequences of particular design decisions, or to compare the performance of a proposed design to alternative systems. Several studies, most of which use text editing as the paradigmatic task, have demonstrated that the GOMS model provides a good description of user behavior and predicts task performance time and learning time with reasonable accuracy (Card, Moran, & Newell, 1983; Polson & Kieras, 1985; Ziegler, Hoppe, & Fahnrich, 1986).

Estimation of Performance Time

The first step in predicting overall performance time for a task is to identify all possible ways in which the task can be achieved, represented by paths through the GOMS model. Each path is defined by statements in the model that are executed when the user follows the path. The execution time for a given path is estimated by summing the times required to execute each individual statement (Card, Moran, & Newell, 1983). The statement times are estimated as follows: one cognitive cycle time per statement plus any Operator time required for statement execution, (e.g., key-hit time, decision-making time), as determined by the analyst.

The overall performance time estimate is the weighted average of individual path times, based on the probabilities of individual path execution during general system use. In the case of the systems modeled here, the overall performance time is the text generation rate, and the individual paths are the different methods used to select letters or words.

Estimation of Learning Time

The empirical formula used to estimate learning time is the sum of 30 minutes for baseline learning time, 30 seconds for each statement in the model, plus any additional memorization time (Kieras, 1987). If two or more statements describe very similar or identical operations, they are counted only once to account for learning transfer gains. Long term memorization time is estimated as 10 sec/chunk of information memorized (Kieras, 1987).

Working memory storage requirements

The GOMS model provides a means of estimating the number of information chunks in working memory at any given time as well as the storage time between retention and retrieval for each chunk. The number of statements that must be executed between retention and retrieval yields an estimate of the necessary storage time for that information (Kieras, 1987).

Alternative Input Systems Modeled

Each of the three computer access interfaces modeled is designed for use by a severely disabled user who can activate only one or two switches. The standard row/column scanning interface consists of a letter matrix that is scanned automatically to allow the user to make a selection using a single switch. The user waits for the system to highlight a particular row, then hits the switch to select the row. The system then highlights successive letters in that row, until the user hits the switch again to select the desired letter. The letters are arranged in order of overall frequency of occurrence (Dabbagh & Damper, 1985), as shown in Figure 1, so that the letters with the highest frequency of use require the fewest number of scan steps for selection. This arrangement stays fixed which simplifies user memorization of letter position. Text is generated by selecting each letter from the letter matrix one by one.

The other two interfaces modeled add word prediction to simple letter scanning in an attempt to improve user performance. These systems exploit the redundancy of the English language in order to predict the user's desired word, thereby reducing the number of physical actions required of the user (Gibler & Childress, 1982). It is assumed that the predictive interfaces use the same letter matrix arrangement described above.

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

FIGURE 1. Standard row/column letter matrix

The first predictive interface studied is a slight variation on the PACA system, developed at Northwestern University (Heckathorne & Leibowitz, 1985). The first two letters of every word are selected using standard single-switch row/column scanning. When the second letter is selected, the letter matrix is replaced by a list of the seven most likely words that start with the two selected letters. If the desired word is not in the first prediction list, the user can select a second prediction list and subsequently choose a word or return to row/column scanning.

The second predictive interface analyzed is the PAL system, developed at the University of Dundee, Scotland (Arnott et al., 1984). The major differences between it and the PACA system are that both its ten-word list and letter matrix are on the screen at the same time, and predictions are made even before a letter is selected and are refined as subsequent letters are selected. If a word is in the prediction list the user hits one switch to initiate one-dimensional scanning of the word list; if not, a second switch initiates row/column scanning of letters.

GOMS Models for the Three Interfaces

Standard row/column scanning. The GOMS model for the standard row/column scanning interface contains seven statements. The only selection path using this interface is a single letter selection from a static two-dimensional letter matrix requiring execution of all seven GOMS statements.

The PACA System. The GOMS model for the PACA system contains 29 statements. There are four possible paths through the PACA system model:

1. Single letter selection for first or second letter of each word.
2. Single letter selection following an unsuccessful search of both prediction lists.
3. Word selection when word is found in first prediction list.
4. Word selection when word is found in second prediction list.

The PAL System. With the PAL system, if the user searches the prediction list after every letter selection, there are only two possible selection paths, as follows:

1. Letter selection after deciding that the desired word is not in the prediction list. (T_1)
2. Word selection when the desired word is found in the prediction list. (T_2)

However, if the word is not present in the prediction lists after the 3rd letter selection, it is assumed that the user does not search the subsequent prediction lists and will select individual letters.

Model Input Parameters for Model Simulation. The first step in comparing system performance times is to establish a set of nominal parameter values to use in the performance prediction equations for each system. The parameters are:

Basic Processor Times	Operators	System Parameters
• cognitive cycle time	• switch hit	• system scan rate
• perceptual cycle time	• word-found	• ave. no. of letters/word
• motor cycle time	• selection-is letter or word	• ave. no. of scans/word selction
	• 1st-or-2nd-letter-of-word	• prediction success parameters
	• at-least-4th-letter-of-word	
	• search list for word	
	• decide if text is complete	

Values for the cognitive, perceptual, and motor Processor times are taken from basic human information processing research (Card, Moran, & Newell, 1983). All three values can be estimated at 0.1 seconds for people without cognitive, perceptual, or motor impairments. These values were used for initial simulation trials as they also represent a wide range of disabled users whose cognitive, perceptual, and motor times (for operating one or two switches) is identical to able-bodied individuals. Time required to hit the switch can be modeled as a simple reaction time, taking one cognitive cycle and one motor cycle, or 0.2 sec.

All except one of the mental Operator times are estimated by determining the relative number of component Processor times. The text-complete Operator is one that cannot be readily subdivided into component Processor cycles. Therefore, a value of 1.35 seconds was used, taken from Card, Moran, and Newell's study (1983) of the generic **M** operator.

The minimum scan rate can be set at the time it takes to perceive a letter on the display and match it to an image of the desired letter plus the switch hit time, or 0.4 sec. Five letters/word was chosen as the nominal estimate for simulation trials (Goodenough-Trepagnier et al., 1982). An estimate of one-half the number of words in the prediction list is used as the nominal value for the number of scan steps/word selection. Overall prediction success parameters for PACA and PAL systems were based on developers' estimates of 70% prediction success (Arnott et al., 1984; Gibler & Childress, 1982).

RESULTS

Performance Time

The results of simulation trials to predict overall text generation rate using the nominal parameters values are shown in Figure 2. The predicted rate for the standard R/C system is 3.58 words/minute (wpm), with the PAL system at 3.16 wpm and the PACA system at 2.92 wpm. These simulation trials predict that the standard R/C scanning system is faster than the predictive interfaces.

Dependence on Number of Letters/Word

Figure 3 shows the predicted text generation rate for each system plotted against the number of letters/word, **L**, when it is varied from 4.5 to 6 and all other parameters are kept at nominal values. The standard R/C system is much more sensitive to changes in **L** than either of the predictive interfaces. This is because the standard R/C system has only one selection path so the number of letters/word is the same as the number of selection loops executed. With the predictive interfaces, a change in **L** affects only those selection loops in which the final letters of a word are individually selected (approximately 30% of the time).

Dependence on Prediction Parameters

The overall proportion of words in the dictionary (70%) can be subdivided into the distribution of words among the prediction lists. For the PACA system, $\mathbf{w}_1$ and $\mathbf{w}_2$ are the probabilities that a word is on the first or second word lists, respectively, given that the word is in the dictionary. Even with $\mathbf{w}_1 = 1.0$, indicating that all words in the dictionary are presented on the first word list, the estimated rate is only 3.01 wpm. For the PAL system, $\mathbf{x}_i$ is the probability of successful word prediction following selection of the **i**th letter. When $\mathbf{x}_1$ and $\mathbf{x}_2$ are varied together from 0.25 to 0.40, the largest estimated rate is 3.22 wpm at $\mathbf{x}_1 = \mathbf{x}_2 = 0.40$.

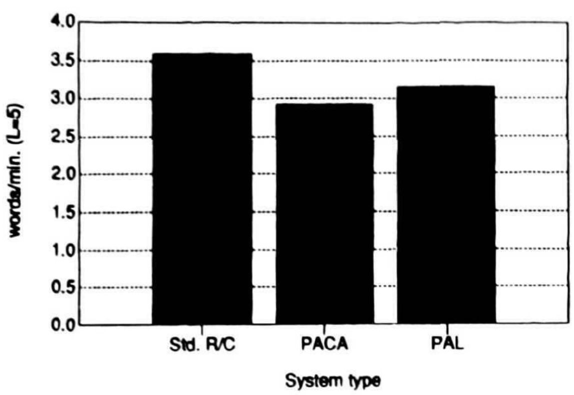

FIGURE 2. Estimated rates using nominal parameters

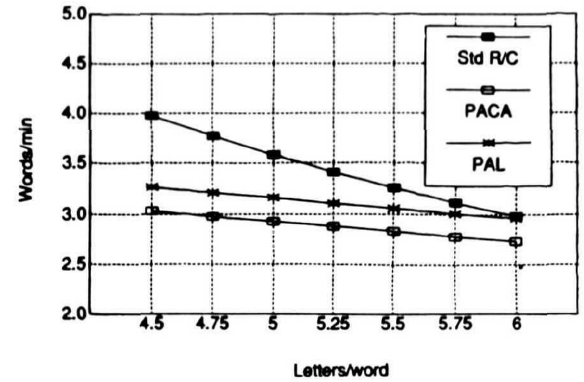

FIGURE 3. Estimated rates as functions of letters per word.

Learning Time Requirements

The GOMS model for the standard row/column scanning interface predicts a basic operational learning time of 33.5 minutes. The time required to memorize the 27 letter locations can be estimated at 22.5 minutes for a total learning time of 56 minutes. The PACA system operational learning time estimate is 43 minutes, and the memorization learning time 23.8 for a total of 66.8 minutes. The PAL system operational learning time estimate is 45.5 minutes, and the memorization learning time 22.5 for a total of 68 minutes.

Working Memory Requirements

None of the systems modeled here places excess demands on working memory capacity or retention time. The largest amount of storage required at any one time is three chunks, which is safely below the five chunk limit suggested by Kieras, and all required retention times are less than one second (Kieras, 1987).

DISCUSSION

Performance Time

Performance time refers to the time it takes to perform the overall task. In the case of an alternative input system, the overall task is to generate text to be spoken in a conversation or used as input to an application program. The ideal case is for the disabled user to approach rates achieved by able-bodied individuals, typically 35 - 40 words/minute for typing and 100 - 200 wpm for speaking. These are unrealistic for a single switch scanning system. The minimum acceptable rate should be above 3 wpm because at rates below this point, conversation breaks down due primarily to the receiver's impatience (Goodenough-Trepagnier et al., 1984). Goodenough-Trepagnier et al. (1984) have shown that receivers' impatience decreases markedly at a rate of 5 wpm, which makes this rate a reasonable target for a minimally acceptable rate.

The preceding GOMS analysis provides estimates of performance time for each system under a variety of conditions. The surprising overall result of this analysis is that none of the three interfaces approaches 5 wpm using nominal parameter values, even though user parameters correspond to those of an able-bodied user. In addition, the two predictive interfaces are at a consistently slower rate than the standard R/C system, with the PAL system somewhat faster than PACA, under almost all conditions.

There are three main factors that contribute to the estimated slowness of the predictive interfaces. First, the number of statements to be executed in use of the predictive interfaces is much greater than the mere 7 statements used in the standard R/C system; this reflects the relative complexity of the predictive systems. Second, use of the predictive systems requires additional mental operators, such as visual search time and word-found matching, which increase the overall text generation rate. Third, the relatively poor predictive ability of the PACA and PAL systems contributes to the slow rate estimate.

The proportion of words present in the dictionary is crucial to text generation speed with predictive interfaces. It should be noted that a major feature of both the PACA and PAL system is that the dictionary contents change dynamically based on the user's word usage. This feature may significantly increase the proportion of words present in the dictionary over time with a resulting increase in text generation rate. This is a user-specific system feature that cannot be easily modeled with the GOMS model. However, the GOMS model can be used to develop criteria for the proportion of words needed to be found in the dictionary in order to achieve a predefined performance level.

Learning

System learning time should be as short as possible, since systems that are difficult to learn will be less acceptable to the target user. Rubinstein and Hersh (1984) propose a "10 minute rule" as a criteria for learning the basics of a system. This may be impossible to achieve as some published estimators of learning time use a base learning time of at least 30 minutes (Polson & Kieras, 1985). A more reasonable design requirement for learning time may be to combine these for a total of 40 minutes.

None of the estimates for the three modeled interfaces meets this design requirement, with the closest being standard R/C scanning at 56 minutes. The two predictive systems have basically the same estimated learning times, at roughly 68 minutes. Note, however, that the estimated learning times include 22.5 minutes for memorization of the 27 letter matrix positions. Memorization of these positions is not essential for use of any system. Therefore, the time required for this memorization can be subtracted from the estimated learning time to give an absolute minimum learning time estimate.

Future Work

This research represents an initial stage in the development of a model that has the potential to become an extremely important tool in the design and prescription of computer access and communication aids for disabled people. The results presented here raise the question as to whether word prediction interfaces, developed as a faster alternative to row/column letter scanning, are actually less efficient than the row/column scanning interface. The model also provides insight into the reasons for this surprising result. First, an overall word prediction success of 70% does not provide enough word selection opportunities to counteract the mental overhead involved in using the more complicated predictive interfaces. Second, when a word is selected, the length of the average word is too short to provide enough switch hit savings.

The quantitative validity of these results is dependent upon the accuracy of the GOMS model descriptions and input parameters. Therefore, one direction for future research is to study the behavior of actual users to determine the validity of the GOMS model predictions.

If previous validation of this approach for analysis of text editing is assumed to carry over to the present application then further sensitivity analysis of input parameters can be expected to yield at least qualitative information about the value of one approach over another. By modeling various techniques common to many different systems, criteria can be developed for system optimization (e.g., determine efficacy of a linear vs. binary search strategy). Future work with the GOMS model is well justified by the potential benefits of an accurate model for human performance with an alternative input scheme for computer operation.

REFERENCES

1. Arnott, J.L., Pickering, J.A., Swiffin, A.L., & Battison, M. (1984). An adaptive and predictive communication aid for the disabled exploits the redundancy in natural language. *Proc 2nd Internat Conf Rehabil Tech*, Washington, D.C.: RESNA, 349-350.
2. Card, S., Moran, T., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Erlbaum Associates.
3. Dabbagh, H.H. & Damper, R.I. (1985). Average selection length and time as predictors of communication rate. *Proc 8th Ann Conf Rehabil Tech*, Washington, D.C.: RESNA, 404-406.
4. Gibler, C.D. & Childress, D.S. (1982). Language anticipation with a computer based scanning communication aid. *Proc IEEE Computer Society Workshop on Computing to Aid the Handicapped*, New York: IEEE, 11-15.
5. Goodenough-Trepagnier, C., Rosen, M.J., & Demsetz, L. (1982). Determinants of rate in communication aids for the non-vocal motor handicapped. *Proc 26th Ann Meet Human Factors Society*, 172-175.
6. Goodenough-Trepagnier, C., Galdieri, B., Rosen, M.J., & Baker, E. (1984). Slow message production rate and receivers' impatience. *Proc 2nd Internat Conf Rehabil Tech*, Washington, D.C.: RESNA, 347-348.
7. Heckathorne, C.W. & Leibowitz, L.J. (1985). PACA: Portable anticipatory communication aid. *Proc 8th Ann Conf Rehabil Tech*, Washington, D.C.: RESNA, 329-331.
8. Kieras, D.E. (1987). *A Guide to GOMS Task Analysis*. University of Michigan.
9. Polson, P.G. & Kieras, D.E. (1985). A quantitative model of the learning and performance of text editing knowledge. *Proc Comp Human Interface Conf*, 207-212.
10. Rowley, B.A. (1987). RPM for accessing large vocabulary files. *Proc 10th Ann Conf Rehabil Tech*. Washington, D.C.: RESNA, 165-167.
11. Rubinstein, R. & Hersh, H. (1984). *The Human Factor*. Bedford, MA: Digital Press.
12. Ziegler, J.E., Hoppe, H.U., & Fahnrich, K.P. (1986). Learning and transfer for text and graphics editing with a direct manipulation interface. *Proc Comp Human Interface Conf*, 72-77.