Med. & Biol. Eng. & Comput., 1981, 19, 725–733

Some considerations in the design of communication aids for the severely physically disabled

P. A. Buhr **R. C. Holte**

Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, R3T 2N2, Canada

Abstract—*This paper summarises some of the design principles adopted during the development of a microcomputer communication aid by the authors (HOLTE, 1978). In such an aid, a graphic representation of selection algorithms gives rise to an objective measure of the efficiency of a communication aid. Existing communication aid designs will be discussed in this new framework. The paper concludes by noting some ways in which a communication aid may be designed to utilise the full capabilities of its intended user.*

Keywords—*Communication aids, Design, Efficiency, Handicapped*

1 Introduction

IN this paper the term 'communication aids' will refer to devices that augment the ability of a communicatively handicapped person to produce communication signals. The paper introduces performance criteria and design techniques for developing a communication aid for the severely physically disabled. The performance criteria will provide a measure of the theoretical speed of a selection method, and this measure will be used to compare selection methods with one another. The design techniques yield a selection algorithm, from user-defined specifications, that attempts to maximise the speed of selection.

The consideration of selection speed is becoming the most important goal in communication-aid design. To achieve maximum speed the communication aid must be tailored to the user's physical and mental abilities. Thus, the communication aid must be designed to achieve maximum speed of selection so that the user can rapidly generate input signals. Then the communication aid must make the most efficient use of these generated signals. The severely physically disabled present a challenge to communication aids designers because of the small number (e.g. three) of discrete input signals that they can generate. The paper will discuss the optimal utilisation of these input signals in terms of speed of selection.

2 Components of a communication aid

Regardless of its physical manifestation, every communication aid can be broken down into a set of fundamental components. Different breakdowns have been suggested in the past (MORASSO *et al.*, 1979; RING, 1980), but we have found the following to be of use in analysing speed of communication (see Fig. 1):

(*a*) Input device (e.g. joystick, pneumatic switch) which translates the user's physical movements into input signals.

(*b*) Output devices (e.g. typewriter, c.r.t. display, speech synthesiser) which displays the selected symbol(s). More than one output device may be available through a single communication aid.

(*c*) Selection algorithm (e.g. linear scan, matrix scan) which uses the input signals to select a symbol from a group of symbols. Henceforth, a symbol is called a selection symbol, collectively called the selection set.

(*d*) Prompting device (e.g. c.r.t. display, cycling light) which reminds the user what symbols are currently available for selection and the actions required to make a selection.

2.1 *Input device*

The input device transduces some type of motor

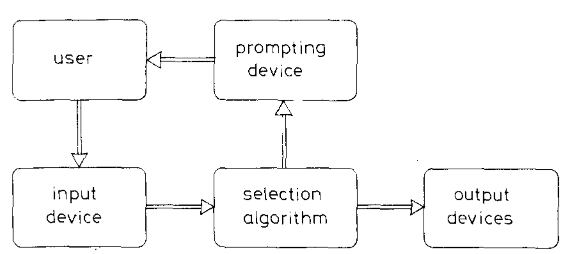

Fig. 1 *Components of a communication aid*

First received 18th September 1980 and in final form 23rd February 1981

0140–0118/81/060725+09 $01·50/0

© IFMBE: 1981

activity by converting it to switch closures which are the input signals to the communication aid. Henceforth, the motor activity that causes an input signal will be called a stroke (simultaneous switch closures can produce a single stroke). A large variety of standard input devices exists, each device is usually customised to fit a particular user's disability.

The importance of the input device in the success of a communication aid cannot be overemphasised. The greatest possible selection speed can be gained through having the most appropriate input device for a user. Unfortunately, it is difficult to formalise input device design to provide maximum input speed given a particular disability. A user and his disability must be assessed together, and the input device configured to utilise those motor skills in which the user is most fluent.

2.2 *Output device*

As symbols are selected, they are displayed on an output device. There are three broad classes of communication output devices: permanent, such as a typewriter; temporary, such as a c.r.t. display; and transient, such as a speech synthesiser. The output devices are the voice and pen of the user; they must satisfy all of his communication needs and they must operate in a medium appropriate to all potential receivers of the message.

A communication aid must be able to deliver an error-free message on behalf of the user. This may involve the use of a correctable temporary output device which is interposed between the selection algorithm and the final permanent output device. Errors in the output can make the meaning of a message difficult or impossible to understand. Also, it can be embarrassing and frustrating to the user to be unable to correct mistakes. Pride in one's work is an important factor to consider. Consequently, error-free output is an important consideration.

If the selection set of the communication aid is extended to include operations which activate appliances (e.g. call buzzer, television, etc.) then the communication aid can double as an environmental controller. Since an environmental controller is often purchased as a separate device with its own power supply, input device, selection algorithm, and prompting device, there is considerable duplication of expense and equipment. The monetary advantage is significant, but the convenience of having only one device—and hence one selection algorithm, one set of input switches, etc.—which serves the dual role of communication aid and environmental controller is the most compelling reason to consider adding appliance actuators to the set of output devices for a communication aid.

Normally, the output device has little bearing on the speed of communication. However, it may affect speed if the output command requires a large time to execute and inhibits selection until it has completed. For example, if a long stored message is displayed the user may have to wait for completion of this task before continuing selection.

2.3 *Selection algorithm*

From the above discussion on input devices, it is evident that all communication aids involve an encoding of the selection set into sequences of switch closures on the input device. The simplest such encoding is a one-to-one correspondence between a switch closure and a message symbol. In this case, each switch can be identified with the symbol it produces (like a typewriter). This is called direct selection.

For a selection set of 40–50 symbols (such as a typewriter), direct selection requires that the user have sufficient dexterity to choose, accurately, one switch from among a large number of switches. In many users, such dexterity is simply not present. In this case, a more elaborate encoding is necessary—one in which there are a small number of switches, and each selection symbol is encoded as a sequence of strokes. Morse code is one such encoding. All of the letters of the alphabet are represented as sequences of the three signals 'dot', 'dash' and 'space'.

Every code can be described in terms of two sets: the set of objects which are encoded (the selection set), and the set of encoding elements (the input signals). In Morse code, letters of the alphabet are the selection symbols and 'dot', 'dash' and 'space' are the input signals. In communication aids, the selection set contains the alphabet and is often extended to include common words and phrases, and punctuation symbols. The input signals of a communication aid are the states of the input device.

From the user's point of view, the encoding of the symbols into his physical movements represents a means for selecting a particular symbol from the selection set. Thus the encoding is called a selection algorithm.

The speed of the selection algorithm is a measure of the efficiency of the encoding of the input signals into the selection set. It will be shown in Section 4 that a quantitive measure can be calculated to express this efficiency and to optimise it.

2.4 *Prompting device*

To enable the user to use a selection algorithm without memorising the encoding it represents, a graphic representation of the selection algorithm is displayed on the prompting device. At any time the user can refer to the prompting device for guidance through the selection process.

The graphic representation of the selection algorithm called the prompting display, must be carefully designed. First, the prompting display must accommodate disabilities specific to the user: for example, visual acuity or scanning idiosyncracies will affect the size and shape of the display. A second, equally important, design consideration is the layout of the selection symbols within the prompting display.

The average selection speed of the user can be greatly reduced because of a disorganised or cluttered prompting display.

In the Section that follows, two prompting displays for the same selection algorithm will be described and analysed in terms of prompting effectiveness.

3 Pictorial representation of selection algorithms

Consider a communication aid whose selection set is comprised of the eight symbols: A, B, C, D, E, F, G, H. Its input device consists of two switches, L and R. The selection algorithm for the aid is given by the code in Fig. 2. The user should not have to memorise this code; therefore, some way must be found of presenting it on the prompting device so that the code for any selection symbol can be found quickly and easily.

One solution is to have a copy of Fig. 2 available to the user at all times. The code for a symbol can be looked up in the table, and the corresponding sequence of motor actions performed to select the symbol. The salient characteristics of this tabular kind of prompting display are:

(i) All selection symbols and codes are visible at all times.

(ii) Once a code is located for a symbol the entirety of the code must be read and translated into the correct switch closures.

to select this letter	strike this sequence of switches
A	L, L, L
B	L, L, R
C	L, R, L
D	L, R, R
E	R, L, L
F	R, L, R
G	R, R, L
H	R, R, R

Fig. 2 *Code for two switches (L & R)*

Most matrix scan communication aids use this display method. (Note, that (i) is not true when overlays are used. More will be said on overlays in Section 6.)

Both of these characteristics may be disadvantageous under certain circumstances. When the number of selection symbols is large (say, 50 or greater), (i) may result in difficulty in locating any particular symbol within the large table of available symbols. (ii) creates problems when the codes become too long to be easily remembered. In this case, the user would have to re-locate the code in the table and remember his position in the code.

To avoid these difficulties, a prompting display must have neither characteristic (i) nor (ii). One such display method will now be described. The prompting display is revised after each stroke, and the only symbols which are displayed at a given instant will be those which are still available for selection at the current stage in the selection algorithm. For example, if on the first stroke the user hits the L switch, then only the symbols A, B, C, D will continue to be displayed. By applying this principle over all possible partial sequence, one arrives at an inversion of the original code (see Fig. 3).

The inverted code gives rise quite directly to a set of prompting displays which can be used to guide the user through the selection algorithm. Figure 4 illustrates the sequence of prompting and user actions involved in selecting the symbol C. Under this prompting strategy, the difficulties associated with characteristic (ii) have completely disappeared; the user never sees the entire code at one time. Characteristic (i), and its associated problems, can be avoided by using a prompt like:

A B C D E F G H < — — ? — — > more letters

in situations where a large number of symbols are available for selection.

3.1 *Inverted code as an N-ary tree*

Representing selection algorithms with their corresponding inverted codes has two positive features:

(*a*) As seen above, the inverted code gives rise quite

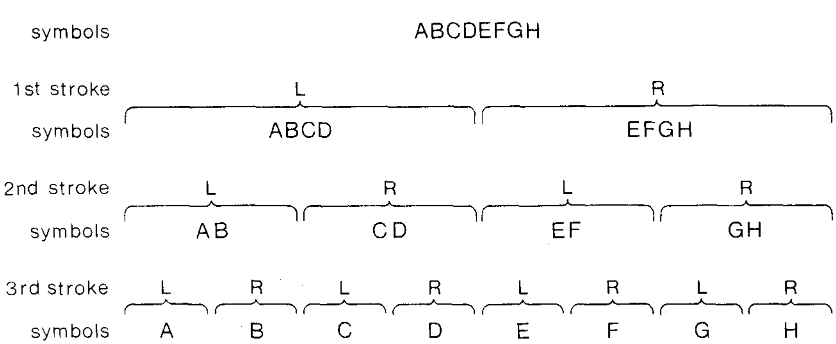

Fig. 3 *Inverted code for two switches (L & R)*

prompt	user action
A B C D ←?→ E F G H	hit switch L (C is on the left)
A B ←?→ C D	hit switch R (C is on the right)
C ←?→ D	hit switch L (C is on the left)

Fig. 4 *Prompt sequence for selecting the letter C*

directly to an effective kind of prompting display.

(*b*) The inverted code may be interpreted as a formal structure called an N-ary tree; thus the theory of N-ary trees can be used for constructing communication aid selection algorithms.

For example, in Section 4, the relative speeds of selection algorithms will be examined using N-ary terminology and results.

The correspondence between the inverted code and the N-ary tree is described below:

3.1.1 *Definition of an N-ary tree.* An N-ary tree is a set of nodes in which each node has N branches, and each branch points to either another N-ary tree (subtree) or a leaf. There is one specially designated node called the root from which the tree begins. The number of branches between the root and a leaf is the rank of that leaf. A balanced tree is a tree in which every branch of any particular node ultimately contains the same number of leaf elements (plus or minus one).

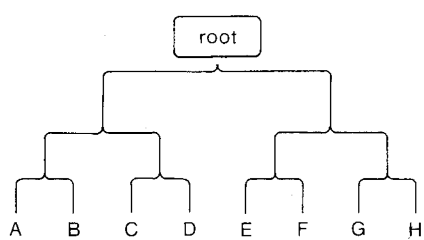

Fig. 5 *Tree representation of inverted code*

3.1.2 *Interpreting the inverted code as an N-ary tree.* The inverted code of Fig. 3 can be interpreted as a tree structure (see Fig. 5). In the tree interpretation of the inverted code, to select the symbol C, one must traverse 3 line segments starting at the root: the left segment, the right segment, and finally the left segment. By relating a line segment to a particular input switch it is possible to translate the action of traversing the tree into the corresponding sequence of switch activations. If the left segment corresponds to switch L, and the right segment corresponds to switch R, then the left segment—right segment—left segment code for C, corresponds precisely to the original L—R—L code for C. This tree is called a binary or 2-ary tree because there are two line segments (branches) extending down from each point in the tree. These two branches correspond to the two switches available for making selections. In general, the selection algorithm for a communication aid with N input functions can be represented by an N-ary tree.

The N branches available at every step in the tree correspond to the N input functions utilised by the selection algorithm. Each leaf accommodates one selection symbol. Performing the selection algorithm is analogous to traversing the corresponding N-ary tree. For example, if a person is using a 4-position joystick, the tree in Fig. 6 can represent the coded sequence of actions for selecting letters. The four branches,

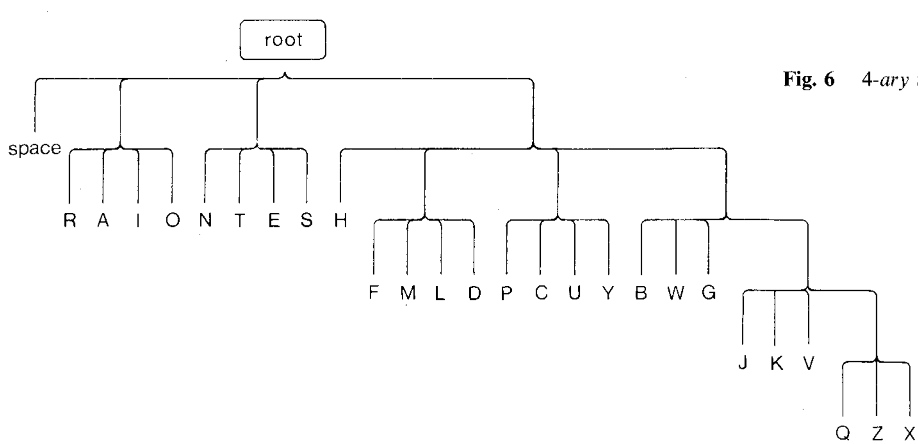

Fig. 6 4-*ary tree selection algorithm*

available at every step in the tree, correspond with the four input functions on the joystick. Each letter corresponds to a symbol available for selection. By moving the joystick, the user can conceptually step from the root along the branches to the desired letter. After choosing a letter, the selection process usually begins again at the root. The prompt segments for each node contain all of the symbols which are beneath it in the tree.

4 Performance measurements for communication aids

For a given user and input device, the average selection speed of a communication aid is determined by the efficiency with which the selection algorithm utilises the inputs, and by the effectiveness of the prompting display. Both the selection algorithm and prompting display are intimately related to an N-ary tree, and their respective desirable properties, efficiency and effectiveness, are also definable in terms of measureable properties of N-ary trees. Thus, it is possible to measure the efficiency of a communication aid; that is, to measure the efficiency of the encoding that exists between the input and output devices.

4.1 *Measuring selection algorithm efficiency*

It is desirable that the user make no wasted motions. The following Sections describe how efficiency can be measured for a selection algorithm:

4.1.1 *Process efficiency.* The average number of strokes required to select objects in the selection set provides a measure of the speed of the selection algorithm. This measure is called process efficiency. For example, if a selection algorithm has an alphabet as the selection set, the process efficiency may be interpreted as the average number of strokes required to select letters while typing common language text. This average can be calculated from the N-ary tree that corresponds to the selection algorithm, and is the sum, over all leaves, of the product of the weight and rank of each leaf. In a tree whose leaves are the letters of an alphabet, the weight of a leaf is the relative frequency of occurrence of that letter.

4.1.2 *Optimal process efficiency.* There are many ways to construct an N-ary tree given a set of elements with known weights. Using Huffman's algorithm (KNUTH, 1973), a tree whose process efficiency is minimal can be constructed. The process efficiency for this optimal tree is called the optimal process efficiency. The selection algorithm corresponding to the optimal tree of letters is that selection algorithm which, on average, requires the fewest strokes to choose each letter.

4.2 *Actual speed of selection*

Optimal process efficiency can be used as a theoretical estimate of the speed of a selection algorithm as follows:

$$\text{selection speed} = \text{u.s.s./o.p.e.}$$

where u.s.s. (user's average stroke speed) is measured in strokes per second, and o.p.e. (optimal process efficiency) is measured in strokes per selection symbol. Selection speed is measured in selection symbols per second. Thus, it is important to minimise o.p.e. to gain the best selection speed.

5 Control functions

Control functions are operations that affect the motion along the branches of a tree and which do not result in selection of symbols.

5.1 *Error recovery strategy*

The simplest error recovery strategy is to have a selection symbol which means 'erase the previous selected symbol'. After an incorrect selection has been made, it can be corrected by selecting the 'erase' symbol and then selecting the correct symbol. The placement of the 'erase' symbol in the N-ary tree can be determined by Huffman's algorithm (using the user's average error rate as a weight), so that the user's typical performance, including errors, will be efficient.

This strategy has a considerable weakness. Suppose the user is traversing a tree and is currently at a point where he must select between one of two subtrees. In error, the user selects the wrong subtree. He immediately recognises his error, however the only means of correcting the error using the strategy above is for the user to continue through the incorrect subtree until he reaches a leaf; then make an improper selection; select the 'erase' symbol; and begin the selection over again.

This recovery strategy requires a considerable amount of skill and understanding to use. In addition to the inconvenience to the user, if the user has a high error rate (typical of beginners) he can compound an error by making another error during an attempt to choose the 'erase' symbol. For these users, a more immediate error recovery technique should be used. This immediate error recovery technique allows the user to make corrections at intermediate nodes during the tree traversal. Two different immediate error recovery techniques are described below:

5.1.1 *Direct reset.* In this strategy, there is a reset control function emanating from every node in the N-ary tree except the root. The action due to a reset control function is to move up the tree to the immediate predecessor of the node whose reset control function was selected. This has the effect of undoing the most recent intermediate step.

The advantage of direct reset is that it allows the user to recover immediately from any intermediate error. This ability is achieved at the expense of process efficiency; for high error rates, the trade-off is justifiable.

5.1.2 *Indirect reset.* Reset control functions are placed in the tree so that although immediate recovery from intermediate-node errors is not possible, it is possible to reach a reset control function from any node. The typical effect of a reset control function in an indirect reset strategy is simply to move the user back to the root; from there he can start the selection procedure anew. For low error rates, the trade-off of more difficult error recovery is offset by a gain in process efficiency.

5.2 *Autoscan*

Autoscan is a control function that selects a particular branch after a period of inactivity on the part of the user. It can cause frustration and a high error rate (STRATFORD *et al.*, 1978), since it takes some control of the device away from the user. On the other hand, it is essentially another input function for the user—no selection becomes a meaningful action.

It is essential that the autoscanning time interval be adjustable. As a user becomes familiar with a selection algorithm the time interval of autoscan can be reduced. This eliminates the frustration created while waiting for the next autoscan. On the other hand, a novice user can be frustrated by an autoscan that speeds past the symbol he wishes to select.

5.3 *Comparison of different selection strategies*

Table 1 gives optimal process efficiency as a function of number of user functions, error recovery scheme, and autoscan/no-autoscan. The selection set is the English alphabet including 'space'. The autoscanning speed is assumed to be equal to u.s.s.

An inspection of the values in Table 1 shows that approximately equal improvements in selection speed will result from:

(*a*) the ability to use autoscan;

(*b*) the ability to manage with indirect error recovery;

(*c*) an increase of one in the number of user functions (because of therapy, or new input device).

However, the amount of improvement is subject to a law of diminishing returns; tremendous gains can be realised by communicators with few input signals, but only marginal increases in selection speed are possible for users with five or more input functions.

6 Analysis of matrix scan communication aids

A matrix scan communication aid has the following features:

(i) The selection symbols are arranged in a rectangle on the prompting device.

(ii) The user traverses the matrix by moving from his current position to an adjacent position in the rectangle.

(iii) In most matrix scan communication aids, the letters of the alphabet are arranged within the matrix according to the frequency of occurrence of the letters in language text, so that the most frequently occurring letters or letter combinations require the fewest strokes on the part of the user (see Fig. 7).

E	T	A	·
N	S	I	·
O	R	H	·
·	·	·	·

Fig. 7 *Matrix of letters*

There are many variations on the matrix scan theme, but the predominant selection algorithm is the row-column scan. In the row-column scan, the user begins at the corner of the matrix. He moves a prompting cursor vertically until he arrives at the row which contains the desired symbol. He then moves the cursor horizontally to the column which contains the desired symbol. After making the selection, the prompting cursor returns to the starting corner.

Table 1. Optimal process efficiency (OPE)

Distinct function N	$(N-1)$ary tree direct reset	N-ary tree indirect reset	N-ary tree direct reset autoscan	$(N+1)$ary tree indirect reset autoscan
2	X	4·094	4·094	2·613
3	4·037	2·613	2·613	2·075
4	2·598	2·075	2·075	1·821
5	2·058	1·821	1·821	1·670
6	1·803	1·670	1·670	1·522
7	1·652	1·522	1·522	1·445
8	1·533	1·445	1·445	1·364

The tree corresponding to the matrix scan selection algorithm is shown in Fig. 8. This is an unbalanced binary tree which requires only two user functions, and could be implemented with autoscan (the right branch) and one user function. When symbols do not have similar frequencies of occurrence (such as an alphabet) the unbalanced nature of the tree results in faster selection. However, when equally probable symbols form the selection set (e.g. numbers) such an unbalanced tree will degrade process efficiency.

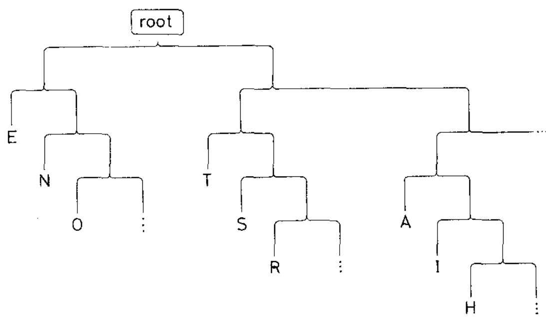

Fig. 8 *Tree representation of row-column scan*

More recent versions of the row-column scan algorithm have employed 'overlays' to counteract the unbalance. For example, if 400 equiprobable symbols are offered for selection, the traditional row-column scan would organise them in a 20 by 20 matrix, and require an average of 20 strokes to select a symbol. Using overlays, the symbols could be presented as four 10 by 10 matrices, in which case the average drops to 12 strokes, a 40% improvement in process efficiency.

The matrix scan communication aids have evolved within the rectangular prompting display paradigm. Although the selection algorithms have improved with time, the matrix format of the display has never changed. This fixed shape, and the rule that one can step only to adjacent squares, results in a highly regular structure which severely limits the kinds of trees which can be implemented. Thus, only certain types of selection algorithms can be used.

For some unbalanced selection sets, in association with a small number of input signals, the matrix scan communication aid is appropriate. For any balanced selection set, and for intermediate numbers of input signals, matrix scan communication aids are inefficient. This is because the matrix scan format precludes the use of more efficient selection algorithms for these cases.

7 Design procedure

The only constraints placed on the designer must be those derived from the user—his abilities and his communication needs. The user's critical characteristics, number of distinct physical motions appropriate as input functions, error rate, visual acuity and scanning ability, mental abilities, motivation, education, etc., must be assessed accurately and used to determine the design of the user's communication aid. In this way, the user's abilities will be utilised for communication in an optimal way.

The constraints imposed by the matrix scan model of selection algorithms appears severe compared with

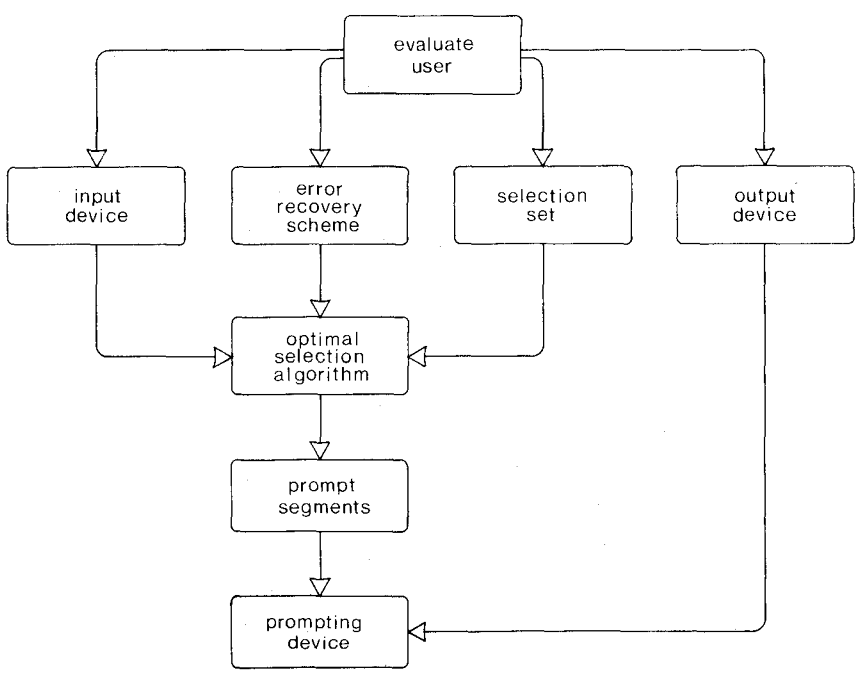

Fig. 9 *Design procedure*

the N-ary tree model of selection algorithms. The overlay variation on the matrix scan technique represents a mild relaxation of the constraints, and results in significant improvements in process efficiency. The N-ary tree model of selection algorithms provides the communication aid designer with more flexibility in matching the user's critical characteristics with the communication aid. The design procedure spawned by this perspective (shown schematically in Fig. 9) is as follows:

(i) Accurate assessment of the user's present physical and mental capabilities is vital. All design specifications will be derived from the evaluation of the user. The effectiveness of the communication aid will be in direct proportion to the precision of the assessment.

(ii) The number and type of input switches will reflect the physical abilities of the user. Only his most fluent motions should be considered.

(iii) Given a set of selection symbols, an error recovery scheme tailored to the user and the number of input functions, a tree of optimal process efficiency can be derived for the user using Huffman's algorithm. This tree gives rise directly to the selection algorithm and indirectly to the prompting segments.

(iv) The output device must satisfy the user's communication needs. Usually more than one output device will be required. An output device that allows the user to correct errors must be available.

The principle is the obvious one—that the device meet the communication needs of the user. But the needs of today and those of tomorrow will differ. User reassessment must be done on a regular basis, and the hardware must be flexible enough that any combinations of its parts—input device, prompting device, selection algorithm, and output device—can be changed with as little cost and delay as possible.

Indeed, such changes should be encouraged by the communication aid. Just as one proceeds from crayon to pencil, to typewriter, to word processor, so the user of a communication aid should move progressively from one selection algorithm to another (the latter a more sophisticated, more effective means of communication than the former) so that he may continue to grow in expressive ability.

8 Conclusion

The main performance criterion for a communication aid is to make maximum use, in terms of speed, of the available input signals a user can provide. Process efficiency provides a theoretical measure of the speed of communication independent of the speed of generation of input signals by measuring the efficiency of the encoding that exists between the input and output device. Thus, process efficiency can be used as a measure for comparison of selection algorithms.

Tree representation of selection algorithms allows the use of Huffman's algorithm to optimise process efficiency and thus selection speed. When this optimised selection algorithm is used in a communication aid it provides a communication aid that imposes no arbitrary restrictions on the speed of communications. The tree representation also gives rise indirectly to an effective prompting display to aid the user during selection.

The tree representation allows a communication aid designer to apply a more objective approach to communication-aid design. In the past, communication aids were often selected for a user on the subjective bases of how well the user fit the communication aid. In an N-ary tree based communication aid the user's abilities are assessed and the communication aid is then constructed to make optimal use of these abilities.

During the design of a communication aid that implemented the ideas in this paper, several other important results were found:

(*a*) The skill of using the prompting segments to traverse the N-ary tree selection algorithm remains constant even if the selection algorithm changes. A new selection algorithm will have a new set of prompting segments but the technique of traversing the new prompting segments is the same as for the old. Hence re-training is minimised, if the user capabilities increase or decrease and the selection algorithm is modified appropriately.

(*b*) Once objects and weights for the objects are generated the construction of the N-ary tree can be automated. This includes the construction of the prompting segments for the prompting device.

(*c*) It is possible to separate the tree traversal algorithm from the tree itself. Hence, it is possible to design a general tree traversal communication aid that can have different N-ary trees inserted dynamically. This allows the communication aid to be changed in time to continually keep up with the changing communication abilities of the user.

Acknowledgments—We would like to thank R. Ellis, J. Muzio, and R. Zarnke for their comments on our paper. Also, we are especially grateful to the Lady's Auxiliary of the Khartum Temple for providing the necessary funding so that we could implement our ideas.

References

HOLTE, R. C., BUHR, P. A., QUANBURY, A. O. and BURKOWSKI, F. J., (1978) *A microcomputer communication aid.* Proceedings of the 7th Canadian Medical and Biological Engineering Conference, Vancouver, 119–120.

KNUTH, D. E. (1973) *Fundamental algorithms.* Vol. 1, 2nd edition, Addison Wesley, 402–404.

MORASSO, P., PENSO, M., SUETTA, G. P. and TAGLIASCO, V. (1979) Towards standardisation of communication and control systems for motor impaired people. *Med. & Biol. Eng.*, **17**, 481–488.

RING, N. (1980) Personal communication. Chailey-Heritage Centre, Lewes, Sussex, England.

STRATFORD, C., ZIMMERMAN, M. E., YOUDIN, M., MILNER, D. and SELL, G. H. (1978) *Environmental and typewriter control systems for use by high level quads: the suitability of various aspects as determined through evaluation.* Proceedings of the 5th Annual Conference on Systems and Devices for the Disabled, Houston, 70–73.

Appendix A

Implementation

A communication aid as described in the paper has been constructed at the University of Manitoba in conjunction with the Rehabilitation Centre for Children in Winnipeg. A commercially available small-system microcomputer was used and the cost of all hardware for the device is comparable with several existing communication aids.

The communication aid supports input devices with one to eight input functions. Any device that can generate the required type and number of signals can be plugged directly into the communication aid. There is also an optional autoscan input function generator. After a fixed period of time an input signal is generated automatically. If enabled, the autoscan function is used in conjunction with the other user generated input functions.

A selection algorithm is stored on a floppy disc. To obtain the desired selection algorithm the appropriate disc is inserted into a disc drive on the front of the communication aid. This reconfigures the communication aid to the new selection algorithm. A library has been started that contains selection algorithms for the different numbers of user input functions, and types of error recovery schemes. The selection set consists of: the English alphabet organised in order of frequency of occurrence; approximately three hundred words organised in alphabetical order; numbers, mathematical symbols, and puntuation characters; and a few environmental controls. The selection set can be extended, reduced, or personalised by creating a new selection algorithm on a floppy disc.

The output devices are a c.r.t. and a line printer. The c.r.t. screen is divided into two sections. The lower portion is used for the prompting segments, and the upper portion is used to display four lines of selected text. The text on the bottom line is correctable and is automatically transferred to the line printer when the bottom line is filled. The line printer can be turned on or off by the user.