COMPUTER COMMUNICATION, ACCESS, AND PROGRAMMING BY SEVERELY HANDICAPPED NON-SPEECH CHILDREN

Dr Harvey A. Cohen
Mr John Monro
Computer Science Department
La Trobe University
Melbourne, Victoria, Australia

ABSTRACT
The La Trobe Talking Communicator is a personal computer adapted to the physical capabilities of the severely disabled. The Talking Communicator offers not only a means of spoken communication and typing capability, with communication capability with other microcomputers. but through programming, in OZNAKI robotics languages, a means for the handicapped user to have a range of concrete operational experiences despite disability. The system whereby a user who can only control one key-switch can optimally control the Talking Communicator is explained. The OZNAKI graphics language Apple][WHAM has been adapted to the l-key user offering exciting creative and artistic possibilities to severely handicapped.

• H.A.Cohen and J. Monro, *Computer Communication, Access and Programming by Severely Handicapped Non-Speech Children*, in R. Welch, Editor, Ninth Australian Computer Conference Schools Symposium 1982, Australian Computer Society, Hobart, pp 237-253 (1982)

Pointer board being used by a non-speech student at the Yooralla Special School, Melbourne. Note that the message "receiver" must give absolute attention to speed communication. There are almost 400 words on the pointer board, and on the other side a letter selector for spelling out other words. A good receiver voices guesses, and accepts a gesture in reply. However for new areas of discourse a pointer board is extraordinarily clumsy. Thus a student absolutely "hooked" on the fireman game described in this paper never told her parents about the experience, and they learnt only through contact with a teacher.

THE PROTOTYPE TALKING COMMUNICATOR

In 1981 a prototype "TALKING COMMUNICATOR" was constructed that enables a handicapped person capable of pressing as few as one but preferably four or five keys to readily communicate with people by speech and by typed text and to conveniently communicate with another computer.

The prototype is based on a 1976 model s-100 bus microcomputer, equipped with 5 inch single density floppy, and is mounted together with keyboard selector, video monitor and speech synthesiser on a wheeled video cart. The keyboard selector is a simple device for swapping between full keyboard for program initiation (and development) and the handicapped user's keyboard. The first models of the prototype incorporated a phoneme synthesiser marketed by Votrax. This synthesiser offers just 63 phonemes, including silences, and some different lengths of the 43 phonemes required for English. Using this phonem synthesiser entails storing in the memory both text and phonetic forms of words/phrases. Subsequently, Votrax have released a single chip phoneme synthesiser, the SC-01 (see Cohen, 1981 a). The current synthesiser, also manufactured by Votrax, incorporates the SC-01 chip, and incorporates a text-to-phoneme translator, with the particular advantage that words composed by letter selection are properly voiced. The synthesiser produces intelligible speech, of a somewhat mechanical male voice.

During the first half of 1982, the project continued to use the prototype Talking Communicator developed in 1981. Hence all photos in this paper show this system. However under construction has been a battery of Mark II Communicators of noteably distinct features. In the Mark II, all RAM is CMOS, which is battery backed. Thus this communicator has inbuilt "mass-storeage", so floppy discs are not required. This makes for a far more compact and portable system, with low power needs, that could be powered by a wheel-chair battery.

• H.A.Cohen and J. Monro, *Computer Communication, Access and Programming by Severely Handicapped Non-Speech Children*, in R. Welch, Editor, Ninth Australian Computer Conference Schools Symposium 1982, Australian Computer Society, Hobart, pp 237-253 (1982)

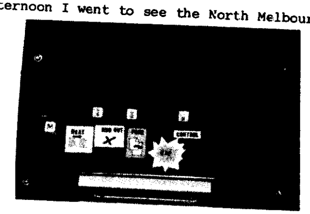
THE UTILITY "TALK"

Central to the Talking Communicator is the utility TALK. In the prototype systems, TALK offered the user 386 words over 8 pages of menu. In TALK, one menu page at a time is displayed on the monitor, above the 4 screen lines in which the statement under construction is composed. Statements, limited to 200 selected items, scroll upwards. Up to 6 different statements can be prepared at a time. The word menus were designed in conjunction with Mrs Leone Phelan, speech therapist at Yooralla. The organisation principle is apparent from the contents of the menu pages 1-8:

Page	Contents
0	Letters, statement select, few phrases
1	When or where it happened
2	Names, occupations, pronouns - who did it
3	Verbs - what was done
4	Adjectives describing feelings, people, things
5	Modifiers eg very, that
6	Places
7	Noun groups: clothes, foods, household objects
8	Miscellaneous

Within the pages 1-8, words are arranged in groups, such as a friends names group, and a foods group. The sequence of pages was designed to minimise the keystrokes needed for selecting in sequence the words of a sentence such as

"On Saturday afternoon I went to see the North Melbourne football match"



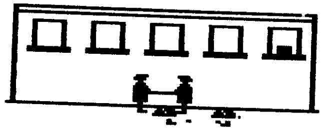
The above photo shows a 5 key keyboard used in the early stages of the project. The keyboard was adapted from an ordinary ASCII keyboard, by shielding out most keys. Only the more accessible key, the space-bar, was used for time-slot selection, this was the ZAP key, which collects the word/phrase/letter adjacent to the cursor. (For this keyboard, there was just the one cursor, which traversed down the columns of the menu matrix.) All the other four keys could be struck at any time independently of the location of the cursor to execute a function as marked:

CCL Next column
DEL Delete last selection
PAGE Next page
CONTROL

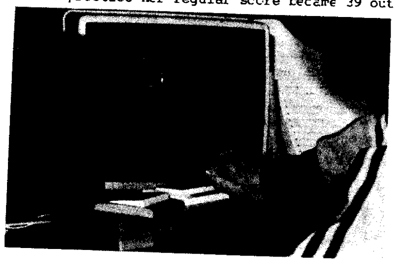
The CONTROL key by itself returned menu page 1. If followed by another key within a time-slot period other functions were formed.

TRAINING OF USERS

Persons with neurological disfunction must be trained to use keyboards. For the five key keyboard, the user has to learn not to strike other than the wanted key, and to do this in a precise manner as speedily as possible. To promote such skills, games are being devised. One such is the Fireman Game.



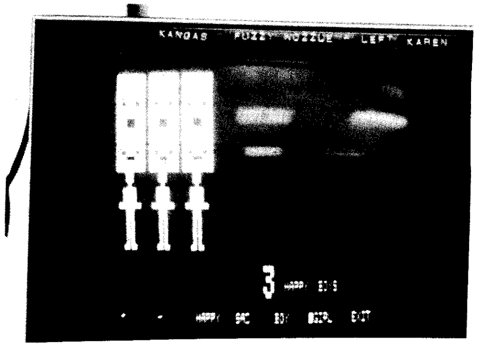
In this game a house with five windows is displayed on the TV screen, and the synthetic voice gives the game rules and advises that the furniture within is being thrown out. For instance there may be a message, "Here comes a chair" and an object appears at a particular window, say number 2, pauses an instant, then slowly falls. If the user strikes key 2 in time, the firemen will rush across and catch the chair, otherwise it crashes onto to ground, with accompanying noise. After 40 moves the game tally is presented. The child pictured below could originally only catch the object 16 times out of 40, but was instantly addicted to this simple game. After two weeks zealous practice her regular score became 39 out of 40.



"Here comes a TV" says the Communicator. The TV has just appeared at window 3. The 5-key trainee user has just struck key number 3. The firemen who were under window 1, amidst wreckage of a chair, are just in the process of moving with their net under window 3, and so are not visible.

• H.A.Cohen and J. Monro, *Computer Communication, Access and Programming by Severely Handicapped Non-Speech Children*, in R. Welch, Editor, Ninth Australian Computer Conference Schools Symposium 1982, Australian Computer Society, Hobart, pp 237-253 (1982)

TIME-SLOT SELECTION FROM A MENU OF COMMANDS: The 1-key PLUSMINUS



THE PLUSMINUS, an elementary number game adapted for control by a handicapped user equipped with just one key. A blob cursor ■ moves along a row of commands one by one, waiting a variable length of time before each command.

If the user strikes a key during the wait period, that command is thereby ZAPPED, otherwise the cursor jumps to the next command in the row.

On the command + one more clown appears

On the command - one clown disappears

On the command HAPPY the clowns start to smile, one by one, and the computer voice counts the clowns one by one as they change facial expression.

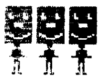
4

On the command SAD the clowns one by one acquire mr Miserable's facial expression, and as eaach clown changes the computer voice count 1,2,3,...

4

On the command BOY the clowns one by one lose their skirts, with the computer voice counting 1,2,3.. until the last clown has become a boy.

On the command GIRL, starting from the left, if the clowns are "boys", then one by one each acquires a skirt, with the computer voice counting 1,2,3...



3.

EXIT this is the command to leave this program.

PROGRAMMING BY THE HANDICAPPED

In order to achieve personal independence, the severely handicapped must be capable of controlling their immediate environment. Firstly, this must include the capacity to turn off and on lights, radios, switch TV channels and the like. This could be very readily achieved using the La Trobe Talking Communicator, by offering the user a page of commands to achieve such effects, and interfacing the electronically controlled switch (through infra-red link perhaps, as is used by some TV remote controllers) to the Communicator. Secondly, there is a need to control a number of robotic devices, such as a book accessor and a feeding robot. Mechanical page turners for books do exist, but these can only advance the pages of a book installed by a parent/teacher, and cannot turn pages backwards. On an experimental basis, a few robotic knives and forks have been constructed. What these future robotic appliances have in common is that they are not just turned off and on, but must be programmed by the user. In order to prepare todays severely handicapped children for the liberating possibilities of personal robotics, it is essential that they gain some experience in programming, especially in robotics. It is fortunate that the robotics languages of the OZNAKI Project (Cohen, 1979) exist, and in the general arena of education have been sharpened and developed: these languages are especially economical of symbols so are very readily adapted to the 1-key user. In OZNAKI, commands are single keystroke, and the syntax is of the brief form of elementary algebra; thus for instance

X=4FR

serves to define X as a new command standing in lieu of doing "F" 4 times, (e.g., F for Forward), and then R just once (e.g., R for Right turn)

There is another vital reason for introducing the handicapped to educational robotics. Piaget has stressed that intellectual development goes through stages, of which the link between earlier ego-centric thinking, and the later "Formal thinking" stage is termed the stage of "concrete operational thought". That is, in this stage, which roughly covers the era of primary/junior high school, the child is very largely developing intellect through experience of manipulation of environment. In the social environment, the child lacking full facility for self expression, such as the non-vocal, can only mature slowly. Likewise, Piaget's studies of intellectual development suggest that the severely physically handicapped will necessarily suffer unless major efforts are made to compensate for their restricted range of manipulative experiences.

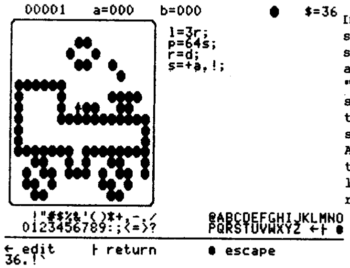
In a very carefully evaluated study (Cohen and Green, 1977) it was shown that normal students using the OZNAKI programming language WHAM showed remarkable enhancement of their spatial abilities, a central component in mathematical problem solving. Handicapped children, being wheel-chair bound, miss out on some of the concrete experiences which Piaget attests are needed for the development of spatial skills. For this reason WHAM has been the first OZNAKI language adapted for use by the severely handicapped 1-key user.

The initial implementation of 1-key WHAM hass been for the Apple][, 48K single disc system. It is planned to also offer WHAM on the Mark II La Trobe Talking Communicator.

• H.A.Cohen and J. Monro, *Computer Communication, Access and Programming by Severely Handicapped Non-Speech Children*, in R. Welch, Editor, Ninth Australian Computer Conference Schools Symposium 1982, Australian Computer Society, Hobart, pp 237-253 (1982)

WHAM

WHAM offers scope for creativity, for creating patterns, solving spatial problems, maze crawling, creating movies, whilst learning basic mathematical ideas such as the role of brackets, the idea of definition, and the elements of programming.



In this printout of the WHAM screen SNAP number 36 has been selected on the command "36." and displayed on the command "!". These keystrokes have been selected by zapping the cursor that patrols the four-group key selector below the NAKI field. A movie of the train is run on the NAKI board on the command p, listed with other macros to the right of the board.

The TV screen during the running of a 1-key version of WHAM is shown above. In the bounded square region on the left of the screen a robot called a NAKI lives. The NAKI takes the form of a flashing arrow. The NAKI territory is like a checker board, with invisible cells, and the NAKI can move from cell to cell. The NAKI movement commands are F for Forwards, which directs the NAKI to move into the adjacent cell in the direction of its arrow heading, and R, for Right turn, which turns the NAKI through 90 degrees. These are the same commands as the LOGO Turtle obeys, but the moves are discrete. As the NAKI roams about the screen it leaves a trail of "dump characters" in the squares through which it passes. There is a choice of several dump characters, and the trail may be invisible. The pattern of marked squares on the NAKI board can be saved on any one of the 64 pages of a library of SNAPS. Once saved, the pattern can be restored to the screen on a single command.

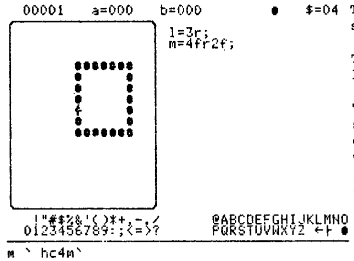
To the right of the NAKI screen is a list of macros defined by the user; these are composed out of basic commands or of other macros. To define macros, one simply selects the slash / command, then writes the definition, concluding with a "return", denoted by ⊢ on the key select. For instance, R is not a basic command in WHAM, but is a macro defined in terms of the basic command D (Direction change) very simply:

R=D

On starting up WHAM the above definition of R is automatically loaded. One can save a set of macro definitions and load them when required.

By showing a sequence of SNAPS one after the other a movie is produced.

WHAM for the 1-key User

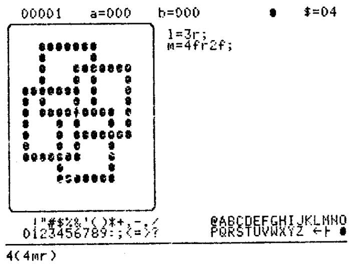


The key selector is below the square NAKI territory.

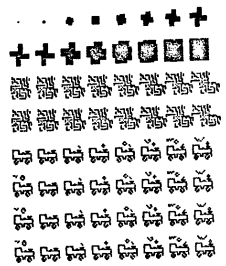
To the right of the territory is a list of macros defined by the user.

To draw this square, the NAKI was sent home (territory centre) on the command h, the screen was cleared with the c command. The macro m, simply composed of f (forawrds) commands, and r (right turn) commands, was repeated 4 times, via the two key-selects "4m" to produce the square.

On the top line of the screen are firstly an iteration counter, then twonumber registers a and b, a display of the current NAKI dump character, and $ the page number at which the SNAP album is opened.

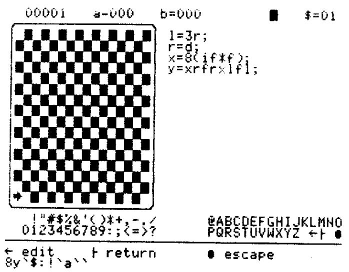


In the second printout of a WHAM screen, the pattern displayed was generated after a home and clear screen command, by the commands shown in the user's command display area at the bottom of the screen, namely "4(4mr))".



Here is a screen print-out of an entire library of SNAPS. The last 32 SNAPS are the frames of a movie of a train, which has rotating wheels and is puffing smoke. Note that each such frame was simply made by modifying the first train snap, then saving on successive $ page numbers. To run a movie requires a simple macro.

• H.A.Cohen and J. Monro, *Computer Communication, Access and Programming by Severely Handicapped Non-Speech Children*, in R. Welch, Editor, Ninth Australian Computer Conference Schools Symposium 1982, Australian Computer Society, Hobart, pp 237-253 (1982)



The territory about which the NAKI roams is like a sixteen by sixteen checkers board, except that the squares, or "cells" of the NAKI board are all the same colour, and not visible unless marked, or occupied by the NAKI. As the NAKI moves about the board, from one cell to another, the vacated cell is marked with one of the various possible dump characters (or none).

The basic move command F, directs the NAKI to jump into the adjacent cell in the direction of its heading. If the NAKI is blocked the command F has no effect, and the NAKI remains where it is. (It is part of the power of OZNAKI robotics languages that there are no "errors", a command that is impossible is simply not performed). This means that a NAKI has a limited "vision" or sense, which can be utilised in programming its activities. In addition to being able to "see" the boundary of its territory, the NAKI can "see" a cell marked with a square blob, ■. The NAKI is forbidden to move outside its territory or to enter any cell marked with a dense blob ■. Hence a pattern of ■'s on the screen may be used as a maze, with the NAKI confined to the other squares.

To display a particular maze, the appropriate album page is selected, then transferred onto the NAKI board by the command "!". At this stage the user can explore the maze by direct guidance of the NAKI. For the tyro, new to WHAM, it is a challenge to use the commands F (Forwards) and R (Right turn) to guide the NAKI through a maze. Incidentally this sort of activity was a major component of the short course in CZNAKI which Cohen and Green (1977) showed with normal children aged 8 to 13 lead to dramatic increases in spatial capabilities.

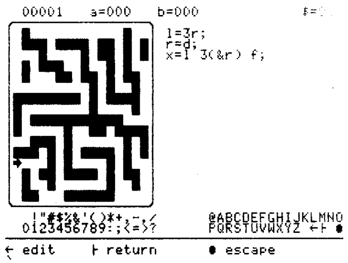
It is fairly easy for someone capable of viewing the whole board to guide the NAKI through a maze. But can one see the board through the "eyes" of the NAKI, and still successfully navigate a maze? This is a problem in robotic navigation. To solve it, one has to imagine oneself as a NAKI, to project oneself onto the TV automata.

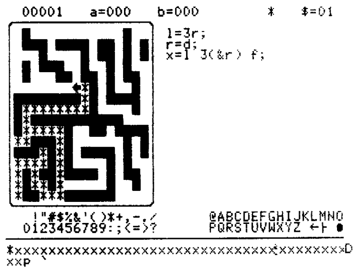
If the NAKI can find its own way out of a maze there must be some command to tell it to do so. The NAKI has just a single "sense" organ, it can detect whether the way forward is blocked. If the way ahead is blocked, the blocked symbol, denoted by a & has value 1, otherwise & has value 0.

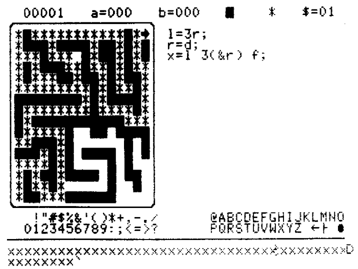
An example of the use of the blocked symbol & is supplied by a macro T:

T=R &T

T is the command for the NAKI to turn right, and then, if blocked to perform the T command. Thus T is the command to turn right, and to keep turning right until unblocked.







The classic way to crawl through a maze is to go forward always keeping one's left hand on the wall. Although this may lead one into many a blind alley, this strategy also leads one out of them, and one never retraces one's steps. For the NAKI to do this, it must be instructed to turn left, then turn right until unblocked, and then go forward. There are several macros that will serve the purpose. Here is one, and the way of writing it. If the macro X which sends a NAKI through a maze ends with a F (forward) command, then the NAKI will go forward if unblocked. If, having moved forward, the NAKI then turns left, it will after no more than three turns be unblocked. In algebraic terms the macro X is:

X=L &F &F &R F

where L has been defined as three right turns in succession:

L=3R

A more sophisticated definition, in terms of the recursively defined T macro given above is

X=2R T F

There are other problems in robot navigation which can be solved by writing simple macros for the NAKI. One such is to circumnavigate an island. This is much the same task as maze crawling. Harder is the task of making the NAKI keep to a set direction, as for instance upwards, past islands. It is easy for the NAKI to get trapped in a bay in the same way as a fly is trapped in a fly trap. In fact the NAKI can emulate several simple types of insect behaviour, such as the way a moth is snared by a candle.

• H.A.Cohen and J. Monro, *Computer Communication, Access and Programming by Severely Handicapped Non-Speech Children*, in R. Welch, Editor, Ninth Australian Computer Conference Schools Symposium 1982, Australian Computer Society, Hobart, pp 237-253 (1982)

THE USER INTERFACE

In designing the hardware and software which allows the handicapped user to access the COMMUNICATOR it was very obvious that the speed of operation would have to be the main consideration.

A good, non-handicapped touch-tyist using the standard "QWERTY" keyboard achieves a rate of sixty words per minute - far short of the 200 wards per minute of normal speech. Improved keyboard designs such as the Dvorjak style promote higher speeds by grouping the most commonly used characters in the "home" position of the fingers but this is of little advantage to a handicapped user typing with one finger. While the layout of the keys on a conventional typwriter style keyboard is of importance, particularly during learning, no mere re-arrangement of keys can achieve sufficient improvement in speed to make this a feasible input device for a handicapped user.

Some of the children we are concerned with can in fact use a modified electric typewriter. The typewriter has a perforated shield covering the keyboard, with the holes arranged so that the desired key can be reached without other keys getting pressed accidentally. A typing speed of about one word per minute has been observed. This speed allows some useful classroom work to be done by the child, but is far too low for use with TALK.

With users of this type the overwhelming problem is lack of hand coordination. They very quickly find the desired key visually, but the process of moving the hand into position, and pressing the key is very lengthy and tedious.

We found that a keyboard with a lesser number of keys was more easily handled. Initially a standard electronic keyboard was used, modified by the removal of all but six of the keys, and by the fitting of a shield with corresponding finger holes. Used in conjunction with automatically scanned tables of characters and words, this greatly improved the rate at which text could be generated. A further improvement resulted when this keyboard was replaced with one constructed using very large rocker-action switches arranged on a 50 cm. arc.

Some of the children however can not use even this input device. A simple yes/no indication is all that can be obtained, using whatever activating method is the most suitable to the particular user. Breath - operated switches, foot switches, and muscle tension operated switches have been investigated. User preference is a good guide here; one girl flatly refused to use her hand operated switch once she had tried a foot- operated switch, and her performance with the new switch was much better.

Once the form of the input device has been decided, the letter selection and word selection/generation strategies can be devised, and the COMMUNICATOR'S programs written accordingly. We have found it convenient to classify users as either "one key" or "five key" types, and this will be used in the following discussion.

TEXT GENERATING STRATEGIES One - key user

For the user capable of only a yes/no response, the most obvious method is to present each character in turn using some sort of electro - mechanical or electronic display, coupled to a typewriter or similar output device. A

character is recorded when a "yes" response is given. Commercial equipment exists which works in this way, but all such methods are inherently slow because of the very long time taken to get to characters near the end of the sequence. When numerals and essential machine - control functions such as "speak" are included, around 40 items must be accessed. The response time of our users is never less than one second, so an unweighted speed of one character per 20 seconds is obtained. (The unweighted speed assumes all characters are equally probable.)

Arranging the most used characters at the start of the sequence helps, but this speed is still far too low for any sort of synthetic voice communication.

This method is analogous to a hypothetical keyboard in which all forty keys are arranged in a line. The user of such a keyboard moves in one space dimension to select a key, just as the user of the sequentially scanned keyboard moves in one time dimension.

Just as there are advantages in arranging a typewriter keyboard in two dimensions, so there are advantages in scanning the characters in two stages : in effe ct providing two "time" dimensions. Example :

> Thirty six characterscan be arranged in an array of six columns and six rows. The rows are scanned by a vertically moving cursor at a rate of about one row per second. When the cursor is opposite the row which contains the desired character the user activates the switch. The cersor then starts moving horozontally across the selected row.
>
> When the cursor is opposite the desired character the switch must be activated a second time. The character is automatically inserted on to the end of the word or sentence being constructed.
>
> The cursor moves back to the origin and commences to scan the rows again in readiness for the second selection.
>
> No character takes more than twelve steps, with the unweighted average being half this.

It can be seen that this "two dimensional" scanning locates one of thirty six characters in 6 steps (average), compared with 18 steps (averege) for the singly scanned method. There is one disadvantage in that twice as many responses are required from the user, but this has not caused any problems.

In view of the improvement in letter selection obtained above, it is interesting to investigate whether further improvements might be obtained by using three or more time dimensions (scanning stages).

Unfortunately the anticipated improvements do not occur for small numbers of items. Example :

> Thirty six characters can be arranged as four 3 by 3 blocks:
>
> The user selects
>
> (1) the block containing the character
> (2) the row containing the character
> (3) the character itself

In this case 10 steps are required at worst (4 + 3 + 3), giving an unweighted average of 5 steps. This is a marginal improvement over the two dimensional scheme, but requires three user responses, and is harder to use.

• H.A.Cohen and J. Monro, *Computer Communication, Access and Programming by Severely Handicapped Non-Speech Children*, in R. Welch, Editor, Ninth Australian Computer Conference Schools Symposium 1982, Australian Computer Society, Hobart, pp 237-253 (1982)

The average time taken to select a character is dramatically reduced if the characters are arranged in order of their probability of occurence in the language used.
For example, the arrangement below requires ONLY 3.1 STEPS ON AVERAGE FOR SELECTING for the letters and space. The time taken to access any other character/function does not significantly change this because they occur far less frequently.

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

It should be noted that the most frequently ccurring characters, that is space (Sp.), E, T, et cetera, occur close to the point at which scanning starts, the top left corner.
At this point it is useful to obtain an estimate of the best speed which is possible in theory. Each step in the scanning process gives the user the opportunity to signal "yes", or, by doing nothing, to indicate "no". Therefore each step represents the the extraction of one binary unit (BIT) of information from the user. It can be seen that at present we are extracting three bits for every letter selected, on average.
It has been estimated by Shannon that English text has an information content of between 0.6 bit and 1.3 bit per character.
It can not be inferred however that we can ever achieve an average of one step per character, selecting one character at a time. It is possible though to devise text generating schemes in which groups of letters, words, or even word sequences are generated. These schemes analyse the preceding text in order to predict the most likely letters to follow, and make these the most easily accessable.
A scheme of this type is being devised to speed up the COMMUNICATOR. As it involves the selection of the first two letters in each word, the strategy used in letter selection is still critical.
It would appear then that there is, at most, room for a factor of three improvement in the speed of TALK, that is we can go from the present three 5 letter words per minute but can not exceed 9 WPM with the one - key user.
The preceding argument assumes a given step time, in this case one second, but in some very severe cases as it can be as long as ten seconds. The response time of the user is dominated by two factors : the decision time and the physical response time.

The decision time can be minimised by making the letter scanning as clear and as simple to use as possible.

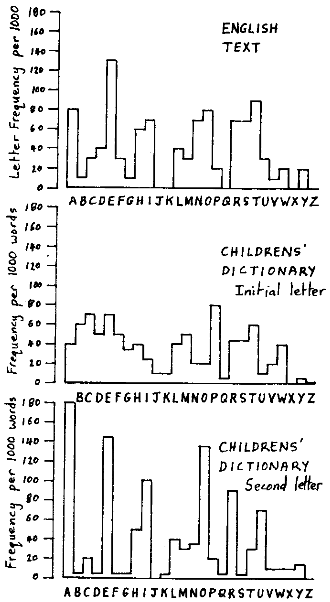
The physical response time can be minimised by giving close attention to the design of the user's switch or switches so that the most appropriate physical action is utilised.

In TALK the step time continually adapts to the medium-term ability of the user. The scanning speed increases gradually until the mistake rate reaches a small but definite value. It slows down if the mistake rate exceeds this. An mistake is detected by the operation of the "rub out" function, or when a row is scanned horozontally without a selection being made.

In this way the user is operating at peak speed, and is neither frustrated by a slow system, nor discouraged by making too many errors.
The five key user has an obvious speed advantage over a one key user. In 5-key TALK characters are presented on the screen in five columns and five separate cursors scan downwards. The user presses the appropriate key when any cursor reaches the desired character. The arrangement used is as

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

This takes on average 1.9 steps to select a letter, compared with 3.1 steps for the 1-key case



In these tables are plotted the frequencies of letters in English text (according to Abramson (1963)), and the frequency of initial letter and of second letter in a 4000 word school dictionary (Manley and Paton, 1981). As can be see seen English text holds a higher proportion of T's than is in this dictionary, because of the high frequency of "the".
By having separate selection tables for the first and second letters in a word the speed improves significantly:
For one calculated system:
2.0 steps for the first letter
1.5 steps for the second letter
Further improvements can be obtained by taking advantage of the highly predictable nature of many letter groups occuriing in English words, and in particular the very strong tendency for consonants to be followed by vowels, and vice versa. However to dynamically change the selection system suggests the use of many tables. But even two selection tables is really too great a burden on the user. There IS a way around this problem. In a version of TALK for the Mark II Communicator currently being implemented an entirely novel approach to scanning is being adopted, which combines an easily learnt table with an adaptive non-sequential scan.

• H.A.Cohen and J. Monro, *Computer Communication, Access and Programming by Severely Handicapped Non-Speech Children*, in R. Welch, Editor, Ninth Australian Computer Conference Schools Symposium 1982, Australian Computer Society, Hobart, pp 237-253 (1982)

THE NON-VOCAL SEVERELY HANDICAPPED

The handicap combination of inability to produce intelligible speech, and poor motor control, is fairly rare, and a highly speculative estimate of the numbers involved would be 8000 in Australia. At the Yooralla Special School, Glenroy, of 130 children with cerebal palsy and other neurological conditions, there are 11 such handicapped children. This disability combination can arise in cerebal palsy (BOBATH 1980,), where brain damage has occured at or before birth, or as a result of brain damage in accidents (which is an increasing phenomena (Ford 1977; Steiner 1981)), or as a result of cerebal haemorage ("stroke"). The victims of certain degenerative neurological conditions also pass through such a stage. A brain lesion need not significantly affect the potential for intelligence, but sadly it can be very difficult to assess the intelligence of the communicatively disabled, some of whom therefore receive no education.

For the non-vocal with good finger control or capable of precise control of a headstick there are a number of communication aids available.

COMPUTER COMMUNICATION, ACCESS AND PROGRAMMING

In 1979 the School's Commission commissioned a survey by Andrews et al (1979) on the needs and priorities of children with handicaps and learning difficulties. Subsequently, the Commission has offered support for a "beacon" project under this writer's direction, titled "Computer Communication Access and Programming by Severely handicapped Children". The broad concern of the project is the needs of children with neurophysical disability that interferes with the normal coordination of muscle action and who consequently have very poor hand control. Many such children cannot speak, or perhaps have very difficult to understand speech. Such children could never write, and use some sort of pointer board, together with limited gestures to communicate. The first task of the project is to provide means for the child to communicate in a classroom situation, both by speech, using synthetic speech, and in hard-copy.

ACKNOWLEDGEMENTS

The staff at the Yooralla Special School, Glenroy, have given the writer, and more importantly the children involved in this project the uttmost suport. We thank especially Mr Keith Grainger (deputy Principal), Mrs Joan Marshal (Physiotherapy Department), and Mrs Leone Phelan (Speech Pathology Department).

Mr Michael Bauer has been concerned with the development of Apple][WHAM. Mr Andrew Hanrah has given valuable part-time assistance in software development.

The work described here had its basis in the support of the Education Research and Development Committee for the OZNAKI project. Since December 1981 the work described herein has been supported by the Schools Commission Project for Severely Handicapped Children.

REFERENCES

ABRAMSON, N. (1963): "Information Theory and Coding", McGraw-Hill New York.

ANDREWS, R.J. (1979): "A Survey of Special Education in Australia; Provisions, needs and priorities in the education of children with handicaps and learning difficulties" Schools Commission, Canberra.

BOBATH, K. (1980): "A neurophysiological Basis for the Treatment of Cerebal Palsy", William Heinemann Medical Books, London.

COHEN, H.A. and GREEN, D.G. (1978): "Evaluation of the Cognitive Goals of OZNAKI: Enhancement of Spatial Projective Abilities", in Wildberger, A.M. and Montanelli, R.G. (editors), "Evaluation of Computers in Education", ACM, SIGCUE, New York, pp 69-90.

COHEN, H.A. (1979): "OZNAKI and beyond", in Harris, D. (Editor), "Proceedings of 1979 National Educational Computing Coference", The University of Iowa, pp 170-178.

COHEN, H.A. (1981): "The development of a TALKING COMMUNICATOR for the Disabled Speechless", Conference Digest, 21st Conference on Physical Science and Engineering in Medicine and Biology, [organised by the Australian College of Physical Scientists in Medicine and other learned societies], School of Medicine, University of Melbourne, August 1981, pp K7.

COHEN, H.A. (1981): "Human Speech and Computer Speech", Conference Proceedings, Second Australian Applied Physics Conference, RMIT, December 1981, Vol 1, pp 217-20.

COHEN, H.A. (1981): "Speech Output for Microprocessor Based Devices: The Designer's Dilemma", IREECON 81 International Conference, (Melbourne August 1981), Proceedings, pp 228-231.

EICHLER, J. (1975): "ETRAN Eye Signalling System", in Participants Resource Book on Non-Vocal Communication System", Cerebal Palsy Communication Group, University of Wisconsin, Madison, 1975.

FORD, B. (1977): "Rehabiliation of the Brain-damaged Patient", Med. J. Aust., 1977, 2: 97-99

GOODENOUGH-TREPAGNIER, C. et al (1981a): "Model for a Computer-Based Procedure to Prescribe an Optimal "Keyboard"", Paper presented at the Fourth Annual Conference on Rehabilitation Engineering, Washington D.C., August 30 - Sept 3 1981

GOODENOUGH-TREPAGNIER, C. (1981a): "Communication Systems for the Nonvocal Based on Frequent Phoneme Sequences", to appear in the Journal of Speech and Hearing Research.

GOODENOUGH-TREPAGNIER, C. et al (1981b): "Derivation of an Efficient Nonvocal Communication System" Preprint. Tufts-New England Medical Center.

HUFFMAN, D.A. (1952): "A Method for the Construction of Minimum Redundancy Codes", Proc IRE, vol 40 no 10 pp 1098-1101.

MANLEY, D. and PATON, J. (1981): "The Piccolo Illustrated Dictionary", Pan Books, London, pp 170-189.

MONTGOMERY, E.B. (1982): "Bringing Manual Input into the 20th Century: New Keyboard Concepts" Computer, IEEE Comp Soc, Vol 15 No 3 pp 11-18.

SHANNON, C.E. (1951): "Prediction and Entropy of Printed English", Bell System Technical Journal, vol 30 no 1 pp 50-64.

• H.A.Cohen and J. Monro, *Computer Communication, Access and Programming by Severely Handicapped Non-Speech Children*, in R. Welch, Editor, Ninth Australian Computer Conference Schools Symposium 1982, Australian Computer Society, Hobart, pp 237-253 (1982)