The Design of a User Interface Management System Which Provides Speech Input for Text Processing

Presented at Speech '88, 7th FASE Symposium, Edinburgh 1988

Technical Report No. 110

Jill Hewitt
Stephen Furner

November 1990

THE DESIGN OF A USER INTERFACE MANAGEMENT SYSTEM WHICH PROVIDES SPEECH INPUT FOR TEXT PROCESSING

Jill Hewitt (1) Stephen Furner(2)

(1) School of Information Science, Hatfield Polytechnic, College Lane, Hatfield, Herts, AL10 9AB
(2) Human Factors Division, British Telecom Research Laboratories, Martlesham Heath, Ipswich, Suffolk

Commercial speech recognition systems are available as 'add-on' units for popular office micro-computers. These systems offer the potential of "hands-free" operation of conventional electronic office products, such as word processors or spread sheets, by the disabled. This paper describes the prototyping carried out to provide access on this type of system to text processing facilities. Initially a "transparent" interface to a conventional word processor was produced. As a result of addressing the user performance characteristics of this system, areas for design improvement were identified and specification criteria for the next version of the prototype developed.

Keywords: Dialogue design, usability engineering, text processing, speech recognition, human factors, engineering psychology, information technology

1. INTRODUCTION

Simple speech recognition systems are now available as 'off-the-shelf' items which can be used as an 'add-on' to a conventional office micro-computer. These system configurations offer the potential of 'hands-free' operation of conventional electronic office products such as word processors or spread sheets. A micro-computer used in this way need not only be employed as a stand alone system, access to a local area network (LAN) and the public telephone network could also be provided as well as the speech recognition capability [6]. In this situation the processing capacity of machines much larger than the original micro-computer could be made available. With network access it becomes possible to consider authoring and distributing documents entirely in electronic form via existing e-mail systems such as JANET or Telecom Gold. Here, the problems of printing or posting the documents would be removed.

These 'add-on' speech recognition systems do not have the same capacity to understand speech that a human listener would possess. Typically, a recogniser of this type would be speaker dependent, require each word to be spoken in isolation, and be restricted in the size of its recognition vocabulary. The speaker dependence would require that all of the items of the recognition vocabulary be retaught to the system every time a different person used it. The difference between the technical performance of a speech recogniser and the way speech is used in ordinary day to day conversation results in significant Human Factors issues arising in its use as an input device [7].

A commercial speech recognition system mounted in a typical office micro-computer was employed to provide a prototype voice controlled text processor for use by the disabled. The recogniser replaced the keyboard to provide "transparent" [2] access to a conventional word processing package. The prototype was provided to a disabled student to use in the production of written work required for a course he was following at university. This indicated a positive attitude towards the system by the user - an average score of +1.3 was obtained using the questionnaire devised by Poulson [5] - and areas for improvement in the design of the user interface were identified.

2.INITIAL PROTOTYPE SYSTEM A "TRANSPARENT" INTERFACE

The Votan Voicekey package was used to build the initial prototype, it provided facilities for alphanumeric characters, or recorded messages, to be produced when an item was recognised by the system. The concept behind the Voicekey software was that it could be used to replace the keys of the computer keyboard. It provided the tools necessary to train the recognition system and associate the material to be produced when a word was recognised. Voicekey ran in background so that it could be used with an application program to provide a link to the speech recogniser. When a word was recognised an alphanumeric string was passed to the application, or a spoken message produced to the user. Some special task words could be trained to carry out special functions, such as displaying the list of words in the active recognition vocabulary.

The initial prototype contained five basic vocabularies which were loaded on the machine when the power was switched on . Every vocabulary was accessible from every other, providing fast switching from one to the other. Subsets of the vocabulary were used to restrict recognition to the vocabulary switching words when a change was required. Text input was carried out with the initial prototype mostly from the "lower" case vocabulary, this contained a phonetic alphabet used for character recognition and some simple cursor movement and editing functions. More extensive editing functions could be obtained from the "writer" vocabulary. On the microcomputer the case shift between upper and lower case was done in hardware on the keyboard. It was not possible to implement a shift character, which could be produced as a result of a successful voice recognition, to shift cases when inputting text. A separate "upper" case vocabulary was provided, although the existence of the "capitalise" command in Perfect Writer meant that switches into the upper case vocabulary were not really necessary. Further details of this initial prototype are given in Hewitt & Furner [3].

3. USABILITY EVALUATION

After the initial prototype had been in use for approximately three months a usability evaluation was carried out with the student who had been using it.

3.1. Evaluation methodology

The evaluation consisted of three performance tests and an attitude measurement procedure.
The performance tests were:-

a) Accuracy of the recognition system - the subject repeated the spoken tokens used to represent the letters of the alphabet, and the digits 1 to 0, ten times each.
b) Text correction task - the subject attempted to correct errors in a prepared text file.
c) Speed test - the subject entered text read from a document for a five minute period using the voice input system and any other method used by the subject for text processing.

The attitude measurement procedure was an application of the technique devised by Poulson [5]. The subject completed a questionnaire and was subsequently interviewed about the attitude profile obtained from the questionnaire. The objective of the procedure was to identify the features of the design of the equipment most influential in producing the scores on the attitude scale. By this method the strengths and weaknesses of the equipment design, from a usability viewpoint, were to be exposed

3.2. Results

3.2.1. Recognition accuracy

a) Alphabet - 92.3%
b) Digits - 97%

These are average scores for all items in the respective vocabularies. There was a large range in the recognition of the items representing the alphabet, the token for w, that was "whisky", was consistently misrecognised for the command "sleep", whereas other items were correctly recognised on all attempts. In 20 cases all 10 recognition attempts for the spoken items were correctly recognised. Thus, for 76.9% of the input alphabet there was not any difficulty of recognition. The letters for which there was a recognition problem in this test with the spoken item used to represent them is shown below.

Letter:	w	i	n	x	a	s
Vocabulary Item:	whisky	i	november	x-ray	alpha	sierra
No: correct recognitions:	0	6	8	8	9	9

3.2.2. Text correction task

The subject was able only to attempt a few of the errors in the text file, and experienced great difficulty with adjusting the text layout to deal with the changes he had made. The errors in the text were too complex to be effectively dealt with by the voice input system. A double spaced layout had been used in the text file in which the subject was to make changes. When a change had been made it required the subject to engage in a large amount of cursor movement, and extensive use of the delete command to remove blank spaces, to reformat the document. This particular type of layout would not be employed in typical documents written with this type of text processor.

3.2.3. Speed test

a) Mouth stick

The subject had a mouth stick and a keyboard adapted for its use. The subject attempted to type with the mouth stick for a five minute period, copying the text from a typed sheet. He was able to input text at approximately 15 words per minute.

b) Text processor

This was abandoned due to recognition errors. Transpositions resulted in cursor repositioning and recognition vocabulary changes. This made text entry very slow due to the time required to recover from these errors. It was far slower than the test with mouth stick.

3.3. Attitude measures

3.3.1. Questionnaire

The average scale value from the questionnaire was +1.3. Three scales resulted in a negative score, these were for "speed of operation" (-1), "need for improvement" (-1), and "discretion of usage" (-0.5). The range of the scale used was from +3 for the most favourable opinion towards the system to -3 for the most derogatory opinion towards the system.

3.3.2. Interview

The subject had a very positive view of the voice input system and saw it as a useful device. He had it implemented on a microcomputer in the room in which he lived. The subject had found that the voice system was slow due to recognition errors, and felt that there was room for improvement in this area. Despite the slow entry speed for text the subject used the system for taking notes whilst reading. This was because he could not operate the keyboard with his mouth stick and use it to turn the pages of the book he was reading from, since the book was in the position the keyboard would normally occupy.

The subject believed that in its current form the voice input system was useful for situations where large amounts of information were being displayed on the computer screen. The example he gave of this was computer based training packages. The voice input system was useful here because it was difficult to see the display screen and operate the keyboard with a mouth stick.

For text processing the subject suggested that the voice system was useful for dealing with situations where small numbers of commands need to be given to deal with large amounts of displayed information. Typically this would be where page layout was being dealt with. For large amounts of text entry the mouth stick was preferred.

The computer was switched on each morning by staff at the accommodation where the subject lived. However, before the subject could use the system he had to find somebody to place the headset microphone on his head. He suggested that a microphone on a flexible stand would increase ease of access to the system giving him greater discretion in its use. With his motorised wheel chair he would be able to position himself at the microphone and thus use the system without a helper.

3.4. Discussion of evaluation results

The voice input system was viewed positively by the subject, and was of use in his studies at the University. The recognition errors reduced the speed of text processing to below that which could be achieved by the subject with a mouth stick. For this reason the subject employed the voice input system in situations where it was not possible to use a mouth stick. This was essentially tasks in which the subject wished to view the computer display screen and issue commands at the same time, or where the mouth stick was required for a secondary task. In this capacity it was important for his studies at University, he used the voice input system to take notes whilst reading from a book. Here, the mouth stick is required for the secondary task of turning the pages of the book.

The misrecognition errors which appeared to result in the greatest difficulty for the subject, were where the cursor was repositioned a long distance from the point in the text where he was working, or where there was an unexpected change in the recognition vocabulary.

4. DESIGN ISSUES FOR IMPROVING EASE OF USE

Following the evaluation of the initial prototype, it was decided to implement some of the recommendations for change using the existing software tool, even though it was apparent that this software tool could not provide all of the required functionality. This enabled a further stage in the 'design - prototype - evaluate - redesign' cycle prior to embarking on the reprogramming of the whole of the voice interface. The major problems which were encountered with the software tool were:-

a) The tool did not allow dynamic voice-controlled retraining of a word nor adjustment of the system parameters to deal with sudden changes in levels of background noise. There was a keyboard facility for this, however it may not always be possible for a disabled user to gain access to it .

b) The tool did not make use of information about the second best matched word to improve recognition accuracy, although the information was available within the underlying system. In the case of a recognition failure it was not possible to enter into a short dialogue to elicit the correct item [4].

c) It was not possible to provide positive feedback that a misrecognition had occurred, for example a warning tone.

d) There was no last resort mechanism available such as stepping through a whole vocabulary and selecting an item using just "yes" and "no". Thus, in a case of severe misrecognition, where voice is the only input modality, the subject has no method for issuing a specific command.

The major advantage of the software tool was that it provided an excellent means of quickly prototyping a system. In the absence of any guidelines on vocabulary structure it was invaluable to be able to use an iterative design approach and enable the user to be actively involved in the design process. For further work on the speech system it is planned to use a more flexible tool produced by VOTAN which allows the recognition process to be controlled using the C programming language.

Four general areas were identified in the initial prototype as requiring modification in order to improve the usability of the device and, thus, to reduce the stress and mental workload encountered by the user, these were:-

a) through a reduction in the cognitive load by employing a simpler model of the vocabulary structure presented to the user.
b) by improvements in the feedback the system provides to the user
c) from enhancing the editing functions to speed up the correction process
d) by reducing the potential for large unrecoverable errors.

5. SPECIFYING A SECOND PROTOTYPE

Rather than use a proprietary word processor the second prototype will employ a text processor specifically written to be used with the speech input system. It is intended however, that a clear separation between the voice input manager and the text processing facilities be maintained within the architecture of the system. The purpose of producing a text processor rather than employing a proprietary design, is to enable more sophisticated error recovery than would otherwise be possible. It is intended to provide error recovery through the use of spoken sub-dialogues and an "undo" command. The "undo" command will be able to deal with command functions within both the text processor and the voice manager. Also, a specialist text processor will enable better use to be made of the feedback provided to the user through the vdu screen. It will be possible to simplify the provision of help, error and information messages through the adoption of a common style of presentation.

The scenario in which the text processor is intended to operate is that of simple e-mail messages. Here, the text editor should provide a robust method for the creation, editing and storage of the messages. It is intended that the editor should provided a vehicle upon which differing strategies for voice input can be implemented in the voice manager and evaluated for usability.

The approach being taken in developing the second prototype is to attempt to explore design options using a formal specification before the system is implemented. This is to allow consideration to be given to alternative dialogue structures and to make sure that there are no loose or dead ends in the design. Specification methods being considered for this activity are SPI [1] and transition networks (see Appendices 1 & 2 for examples). Although state transition networks provide an easily understandable visual record of a dialogue, SPI provides constructs for the simple expression of parallel activities. The parallel construct in SPI makes it possible to maintain the separation between the text processor and voice input management in a fairly straight forward manner.

Two levels of description are provided within SPI these are called EventCSP and EventISL. An EventCSP specification provides a high level description of the semantic structure of the dialogue, and the EventISL gives substance to the elements defined in the EventCSP description. The SPI notation has been devised specifically for the representation of human computer interaction in a parallel event based environment, and, as such, it simplifies the problems of integrating the speech input with other facilities at the interface. Apart from providing a rigorous description of the activity at the interface, the notation can be exercised upon on a computer to examine the adequacy of the design being specified.

6. CONCLUSIONS

This work has shown that it is feasible to use simple speech recognition for text processing by spelling out the words a letter at a time. While this is too slow for commercial text processing it can provide a viable means of interacting with a micro-computer for users with restricted keyboard access. Advanced telecommunications facilities are becoming more widely available through technological developments such as the integrated services digital network (ISDN), and the proposed European integrated broadband communications network (IBCN). The ability to interact effectively through terminal systems is a significant issue if the disabled are to be able to take advantage of the opportunities presented by these developments in information technology. Before facilities such as electronic mail systems, or public bulletin boards, can be used it is first necessary to be able to create text and control the terminal system.

References

[1] H Alexander, "Formally-based tools and techniques for human-computer dialogues, Report TR.35 Dept. Comp. Sci. University of Stirling, 1986

[2] H H Dabbagh, R I Damper, & D P Guy, ("Transparent interfacing of speech recognisers to microcomputers", Microprocessors and Microsystems,(1986), 10, 371-376.

[3] J Hewitt and S Furner, "Text processing by speech: dialogue design and usability issues in the provision of a system for disabled users, Proc. HCI'88 - People and computers, Manchester (UMIST), 1988

[4] P C Millar, I R Cameron, & D J Chaplin, "A robust dialogue control strategy to improve the performance of voice interactive systems", Proc. European Conf. on Speech Technology (1987).

[5] D F Poulson, "Towards simple indices of the perceived quality of software interfaces", IEE Colloquium Digest 1987/38 "Evaluation techniques for interactive system design 1"

[6] J Schnabl, & R Boissinot, "A voice controlled integrated communication workstation", Speech Technology, (Oct/Nov 1987).

[7] C A Simpson, M E McCauley, E F Roland, J C Ruth, & B H Williges, "System design for speech recognition and generation", Human Factors, (1985) 27, 2

Acknowledgment is made to the Director of Research and Technology of British Telecom for permission to publish this paper. Also, to BT Action for Disabled Customers for their assistance in support of this work.

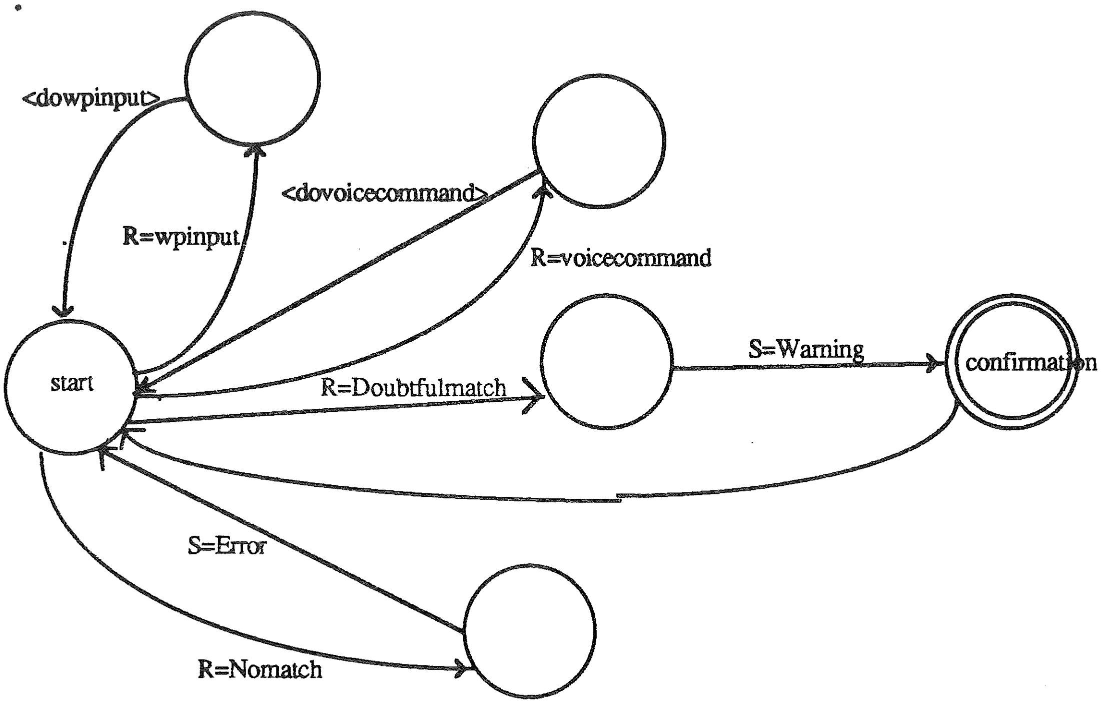
UIMS FOR TEXT PROCESSING BY SPEECH

Appendix 1
Specification example using Event CSP

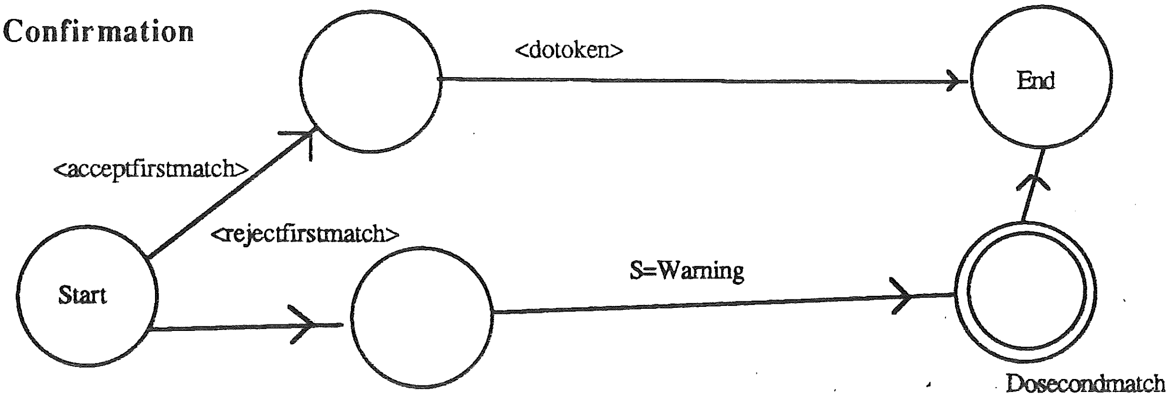
```
speech-operated wp = voiceint || wp

voiceint = (recognisewpinput -> dowpinput ; voiceint
           [] recognisevoicecommand ->dovoicecommand ; voiceint
           [] doubtfulmatch->warningsound->confirmation ; voiceint
           [] nomatch->errorsound->voiceint)

dovoicecommand = (undo->doundo
                 [] repeat->dorepeat
                 [] showkeys->recogniseanything -> removekeys
                 [] start
                 [] stop->savestate->suspend
                 [] train->savestate->dotraining
                 [] parameters->changeparameters
                 [] switch->modechangeindicator->doswitch)

doundo = (validwpundo
         []  invalidwpundo -> errorsound
         []  validvoicecommandundo
         []  invalidvoicecommandundo -> errorsound)

dorepeat = (recognisenumber -> repeatprevioustoken
           [] doubtfulmatch -> errorsound
           [] nomatch -> errorsound )

repeatprevioustoken = (dotoken->(again? -> repeatprevioustoken
                                [] nomore))

confirmation = (acceptfirstmatch -> dotoken
               []  rejectfirstmatch -> warningsound -> dosecondmatch)

dosecondmatch = (secondmatchavailable -> (accept2ndmatch -> dotoken
                                         [] reject2ndmatch -> errorsound)
                [] secondmatchnotavailable -> errorsound )

dotoken = (dowpinput
          [] dovoicecommand)

suspend =( recognisestart -> voiceint)

wp = (awaitinput -> (dokeystrokes
                    [] tryundo
                    [] wpmenu)
```

Note: terminal symbols are underlined

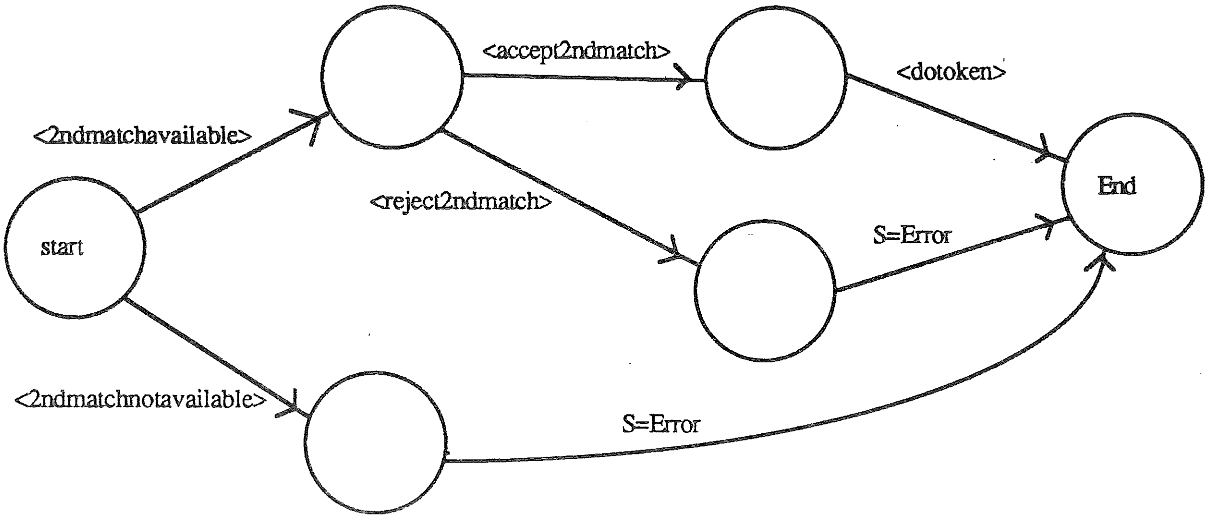
Voice Input



Confirmation



Dosecondmatch



<accept2ndmatch>
<dotoken>
<2ndmatchavailable>
End
<reject2ndmatch>
start
S=Error
<2ndmatchnotavailable>
S=Error