MODELLING OF SPEECH-BASED USER INTERFACES

Brian Mellor
Speech Research Unit,
DERA Malvern,
St Andrews Road,
Malvern, WR14 3PS,
England.

Chris Baber
School of Mechanical Engineering,
University of Birmingham,
Edgbaston,
Birmingham, B15 2TT,
England.

ABSTRACT

The capability profiles of commercial automatic speech recognition (ASR) systems are rapidly improving in terms of vocabulary size, noise robustness and user population. Most contemporary applications of ASR use interfaces relying solely on the speech mode of interaction (over telephone channels for example). Many applications will, however, benefit from using speech input in conjunction with other interaction devices such as trackballs, keyboards and touch-screens.

In this paper, we present an interface modelling approach based on a critical path analysis of the interface design. The approach has been developed to model multi-modal interactions using combinations of input devices. Degradation of unit performances allow the effects of environmental factors on the overall interface performance to be predicted.

The model is verified by comparison with experimental trials carried out on a number of multi-modal applications. It is demonstrated that the model is able to predict the main performance metric (task completion time) to within 10% of the experimental values.

1. INTRODUCTION

Automatic speech recognition (ASR) is an ever-maturing technology with continual improvements to system capability profiles. Larger vocabularies and speaker populations, low enrollment and improved environmental robustness facilitate an increasing number of commercially viable applications. Large vocabulary dictation systems are offering connected word performance, whilst there are increasing numbers of small vocabulary telephone based services. The applications of medium-sized vocabulary recognisers are also becoming numerous [Fraser and Dalsgaard, 1996].

There still remain, however, some problems with the use of ASR in real-world and laboratory applications. Recent papers by [Basson, 1996] and [Isobe et al, 1996] report disappointing performance in telephone based systems compared to touch-tone interaction. Reports by [Damper, 1995] and [Mellor et al, 1996] suggest that trackballs and keyboards out-perform speech input devices in laboratory trials. It is, therefore, important to understand why ASR is failing in these applications: is ASR an inherently less able interface modality or is ASR being implemented inappropriately in interface designs ?

Many ASR applications rely solely on the speech modality, due either to the restricted communications medium (over telephone channels for example). There are, however, many applications where speech is best used in conjunction with other input modalities. In voice dictation tools, speech recognition provide input to the primary text creation task. Speech recognition in a moving vehicle to control a secondary task will inevitably compete for user resources.

Spoken dialogue interfaces reported in the open literature (at ISSD-96 and ESCA-NATO Workshop on Spoken Dialogue for example) are commonly developed by a 'craft' approach based around 'Wizard-of-Oz' trials and prototyping. There is a requirement to supplement these approaches with an interface modelling tool based on measured unit performances (such as the timing and accuracy of key-presses). Such a model would allow rapid identification of unsuitable applications and designs.

In this paper, we describe such a modelling tool based on a critical path analysis (CPA) approach. Unit performance values for interface devices are taken from the open literature and from controlled experiments. The model described here is able to describe interfaces where several modes of interaction take place simultaneously. In addition, by estimating performance degradation of individual atomic actions, the model is able to predict the overall effects on the man-machine interface of environmental factors (stress, workload, vibration etc.).

To verify that the CPA approach to modelling is valid and to 'calibrate' the model parameters, a number of interface designs of varying complexity have been developed and trialled. The applications include alpha-numeric data entry, a simulated surveillance task and electronic map manipulation.

2. THE INTERFACE MODEL

Task Transaction Time

Although it may be of interest to create predictive models based on the use of 'knowledge' in the interaction, validation is difficult as cognitive processes are largely hidden. Increased error rates, workload and efficacy of the interface design will all observably affect the transaction time. There are drawbacks with the use of this metric when it comes to trying to apply timing values to unseen, cognitive processes. However, by designing tasks which are assumed to contain various levels of cognitive processing allows extrapolation of a processing 'time'. The interface model we require is

thus one which can combine unit times into an overall task time.

Model Requirements

Given that some interaction devices require the same human resources (mental, physical or both) the modelling environment needs to be able to model conflict and competition in the interface. Conversely, the model needs to explain and predict co-operation in the input, where modalities are used in conjunction ('put that there'). We require that the model can predict task transaction time to within 10% of experimentally derived values. The model also needs to include a description of environmental factors - noise will reduce ASR performance, vibration will reduce keyboard performance etc.

Some previously reported interface models, for example [Card et al, 1983] base their analysis on a 'one-best-way' approach where the multiple input devices are utilised in a prescribed fashion. This does not allow for the flexibility in the human use of an interface. It is desirable to limit such constraints of the model so that various interaction strategies can be described. To predict transaction times requires an approach where unit times for discrete interaction events can be combined. Finally, it would be advantageous to use the model to analyse existing application designs to identify flaws.

Critical Path Analysis

Critical path analysis is a task scheduling technique traditionally associated with project management. The use of CPA to model human performance was suggested by [Schweickert, 1980] to model transaction times in choice reaction tasks. The model we have developed extends this approach to cover multi-modal interactions. The analysis is based on sub-task transaction time and is ideally suited for modelling time ordered events with degrees of dependency and parallel tasks. The critical path through the interaction is identified from a description of the sub-task transaction times and knowledge of the interface design. The time taken to complete actions on the critical path provides predictions of the overall transaction time. By analysing the steps on the path, it is possible to analyse the interaction in some detail.

Use of the Model

Using the modelling approach is a three stage process. The first stage develops a detailed task description, starting with a high level task description of the form:

DO(task) WITH (modality) USING (device) WHILE (other tasks) IN (environment)

For the case of the simulated surveillance task described in Section 4, the top level description for speech input can be written:

DO[target classification]WITH[speech]USING[ASR]
WHILE
[DO[mental arithmetic]
WITH[verbal/auditory]USING[speech]]
IN[recording booth]

By analysing the modes of interaction identified in the high level description, it is possible to draw up a conflict matrix which describes how the various modes of interaction will interact. Table 1. shows a mode conflict table based on an implementation of multiple resource theory [Wickens, 1992].

Code	Input	Processing	Output
VM	•acquire target	•classify target	
VA	•receive sum	•mental arithmetic •generate report	•position cursor •data entry •speak response

Table 1. Conflict table for input modes. VM-visual/manual, VA- verbal/auditory modalities.

From Table 1. it is predicted that the main conflicts will occur using the verbal/auditory modality for processing the mental arithmetic task and generating the report. In addition, conflict is expected in the output between speaking the mathematics task result and operating the speech recogniser.

The final description of the interface design can be carried out using hierarchical task analysis (HTA), introducing specific features of the interface devices. Although HTA is traditionally used to explore existing machine interfaces, it allows users to consider the interface design in more detail.

The second step is to identify the temporal relationships between the operations described in the task description. These dependencies and the conflicts identified in the conflict matrix can then be used to implement the CPA model of the interface. The CPA network is then

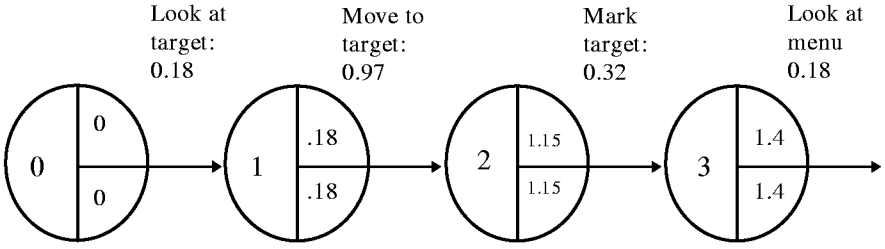

Figure 1. Partial CPA diagram for surveillance task

developed to model the task dependencies and parallel working identified in the HTA description.

Using the CPA model in a forward direction provides predictions of the time to completion of the desired interaction. If experimental results are obtained, these can be used to constrain a backwards pass using the model to identify various task performances in the interaction. Figure 1. shows a segment of the CPA model which describes target selection for the surveillance task detailed in Section 4. The labelled nodes (0,1,2 etc.) contain information on the earliest and latest start times for the operation. The links contain data on unit times for the labelled operation.

3. EXPERIMENTAL TRIALS:

The accuracy and validity of the model was verified against several interface designs of varying complexity. The interface designs were implemented using the 'GUIDE' rapid prototyping toolkit as reported in [B. Mellor et al, 1995]. Speech recognition performance using the 'AURIX' reconfigurable speech recogniser was sufficient to allow its use rather than a 'Wizard-of-Oz' simulation. The performance of the rapid prototyping model set used would not have produced the performance expected from speaker and task-dependent models.

Registration Number Entry

Initial calibration of the model was carried out on experimental data reported in [Mellor and O'Connor, 1995]. This trial interface required the transcription of British vehicle registration numbers using either ASR or standard keyboard. A secondary control was provided to vary the presentation rate, using keyboard control for ASR transcription and ASR for keyboard entry. For voice data entry, the recognition vocabulary comprised the ICAO alphabet ("alpha", "bravo", "charlie" etc.), plus the digits "zero" to "nine". For keyboard entry, individual key presses were used. No error correction was available for either input modality.

Values used for the model assumed data entry of 350 milliseconds per word at 85% word accuracy. An extra float was used to model the time taken to translate alphabetic characters into ICAO vocabulary. The float times were 1.25 seconds for speech input and 1.6 seconds for keyboard input. Initial delay times were derived from a backwards pass through the model and relate to system response time, user preparation and so forth. These delays were 3.16 seconds for keyboard entry and 2.26 seconds for speech input indicating an interaction between the task performance and interaction device.

Table 2 shows a comparison of the experimentally derived task completion times and the model predictions for speech entry. Speech recogniser errors were modelled as repetition of input characters, lower accuracy requiring more repetitions and hence longer transaction times.

Transcription device	Predicted task time (secs)	Actual task time (secs)
ASR	125.6	126±14
keyboard	115.2	115±21

Table 2. Model predictions for ASR entry of registration numbers

Simulated Surveillance Task

The second trial required participants to select one of four target objects displayed on a computer screen and then complete pro-forma report on the object's properties. Input modalities used were ASR, miniature trackball and a cursor keypad, as reported in [Mellor et al, 1996]. In addition, a mental workload task was given by way of an aurally presented arithmetic task, at three levels of complexity.

Recognition errors were corrected by repetition. This is represented in the model as an increase in the number of commands required for any action. Table 3 shows the predicted task completion times for various ASR word accuracies. The task completion time averaged over all subjects was 34.7 ± 11 seconds, with an average recogniser word accuracy of 74 ± 20 %

ASR Accuracy (%w/a)	Predicted completion time (secs)
60	31.8
70	31.1
80	30.4
90	29.0

Table 3. CPA prediction of task completion time for ASR input for differing ASR performance.

Workload	Predicted ASR time (secs)	Actual ASR time (secs)	Predicted trackball time (secs)	Actual trackball time (secs)
No Work	12.23	12 ± 12	8.36	8 ± 2
Medium Work	15.06	15 ± 9	9.64	11 ± 5
High Work	16.28	16 ± 9	10.87	13 ± 5

Table 4. Predicted completion times for the report filling task for ASR and trackball inputs under workload.

Workload Condition	ASR % increase	Trackball % Increase
Low Work	100	100
Medium Work	123	115
High Work	133	130

Table 5. Modifications to unit transaction time due to workload. Values extracted from the backwards pass.

The effect of workload in this application was modelled by increasing the time taken for individual interface actions. These values were extracted from the backwards pass through the model. Using the modified

timings, we get the figures in table 4 which show the predicted and actual task completion times under the various workload conditions. Table 5 presents the modifications to the unit transaction times used for this modelling.

Map Manipulation

In the final reported trial, the experiment required the manipulation of an electronic map using either a cursor keypad, ASR or a miniature trackball. The scenario was the manipulation of a digital map in a small hand-held tablet computer. Participants were tasked with 'marking' specified features, navigation instructions being provided for each target. Two types of task were implemented - marking of a feature some way off the display (requiring control of scrolling functions) and marking of feature on the visible display.

Table 6. shows the comparison of the predicted transaction time with the average transaction time from the experimental trials for all the input devices.

	Predicted (secs)	Actual (secs)
ASR	312.7	320
Keypad	225.7	224
Trackball	370.4	370

Table 6. Predicted and actual task completion times for map manipulation trials.

4. DISCUSSION

The predicted transaction times above appear to match well with the experimentally derived results. To examine whether this agreement is statistically significant, we followed the analysis of [Liao and Milgram,1991]. We limit the expectations of the model to predict the transaction time to a 10% error. It is not sufficient to examine the null hypothesis that the predictions and results are from the same distribution due to the allowable error. We thus want to examine the likelihood of a false rejection of the hypothesis rather than a correct acceptance.

Using this criterion with the surveillance task data, the model predicts the experimental results to within 10% of the experimental measurements to an 80% confidence level. For the map manipulation task, there were insufficient experimental trials to examine agreement to this level of confidence. However, the results do suggest that the model is providing predictions within the acceptable range of results.

5. CONCLUSION

Automatic speech recognition has traditionally been used in isolation, either over a telephone channel or as the primary mode of interaction (for voice dictation tools). ASR will increasingly be used in conjunction with other control devices and activities, in multi-modal interfaces or to control secondary equipment (such as navigation systems in vehicles). We have presented an approach which aims to model such complex interfaces using a critical path analysis technique.

The interface design is analysed though task description formalisms and application of hierarchical task analysis to identify dependencies and resource conflicts. The resulting design is implemented as a CPA network, with unit times being taken from empirical values. The overall transaction time is thus simply the sum of the unit transaction times for the processes on the critical path. There is some evidence that the model predictions agree with the results from experimental trials. More observations are still required to increase the confidence in this result.

By implementing a backwards pass through the model with empirically derived times, it is possible to examine the interface design in great detail. This will provide insights into the way users interact with the applications and hence lead to design improvements.

6. REFERENCES

Baber, C. and Mellor, B. 1996.. *Modelling Transaction Time for Dual-Tasks Using Critical Path Analysis.* Proc. 1st. Intl. Conf. Engineering Psychology and Cognitive Ergonomics. Stratford-Upon-Avon.

Basson, S., Leung, H. and Pitrell, J. 1996 *User Participation and Compliance in Speech Automated Telecommunications Applications.* Proc. ICSLP-96, Philadelphia.

Card, S.K. et al. 1983. The Psychology of Human-Computer Interaction. Hillsdale, NJ:LEA.

Damper, R., Tranchant, M. and Wood, S. 1995. *Speech Versus Keying in the Human-Computer Interface.* Proc. ESCA-NATO Workshop on Spoken Dialogue Systems, Vigso.

Fraser, N. and Dalsgaard, P. 1996. *Spoken Dialogue Systems: A European Perspective.* Proc ISSD-96, Philadelphia.

Isobe, T. et al. 1996. *Voice Activated Home Banking System and Its Field Trial.* Proc. ICSLP-96, Philadelphia.

Mellor, B. and O'Connor, C. 1995 *User Adaptation to Voice Input Interfaces.* Proc. ESCA-NATO Workshop on Spoken Dialogue Interfaces, Vigso.

Mellor, B., Tunley, C. and Baber, C. 1996. *Evaluation of Automatic Speech Recognition as a Component of a Multi-Input Device Human-Computer Interface.* Proc. ICSLP-96, Philadelphia.

Kiao, J and Milgram, P. 1991. *On Validating Human Performance Simulation Models.* Proc Human Factors Soc. 35th Annual Meeting, Santa Monica CA.

Schweickert, R., 1980. *Critical Path Scheduling of Mental Processes in a Dual Task.* Science #209.

Wickens, C.D. 1992 *Engineering Psychology and Human Performance* (2nd edition.), New York, Harper Collins.