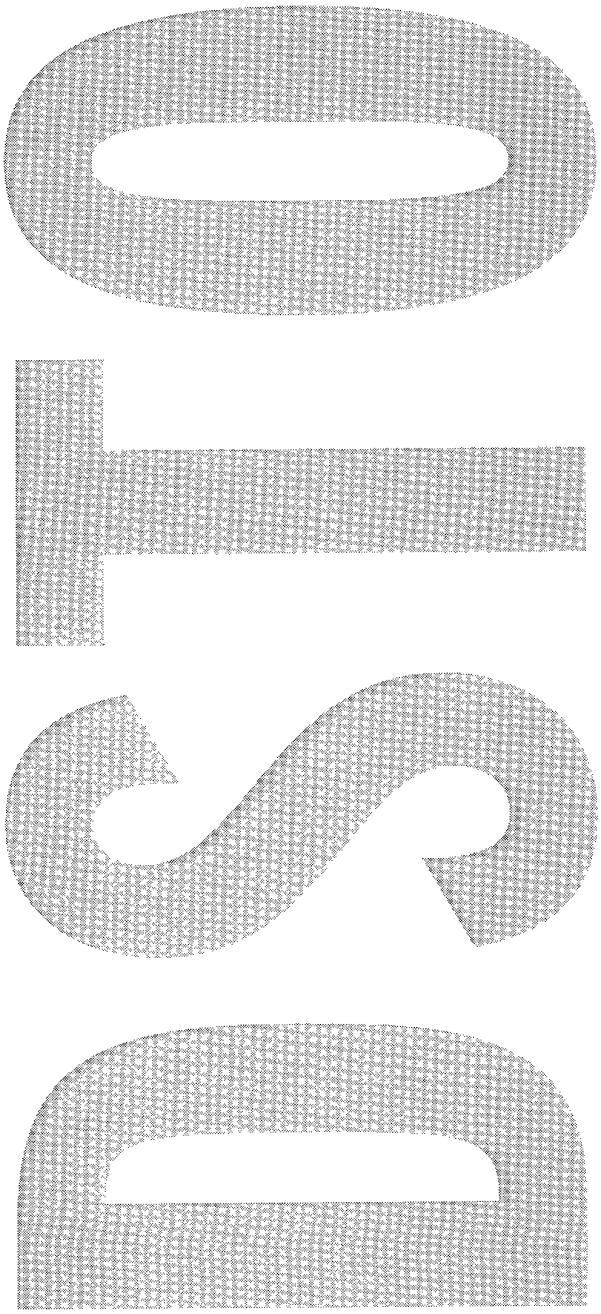
DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO



Experimental Comparisons of Data Entry by Automatic Speech Recognition, Keyboard, and Mouse

Helen Mitchard and Jim Winkles

DSTO-RR-0220

Experimental Comparisons of Data Entry by Automatic Speech Recognition, Keyboard, and Mouse

Helen Mitchard and Jim Winkles

Information Technology Division
Electronics and Surveillance Research Laboratory

DSTO-RR-0220

ABSTRACT

The objective was to determine the conditions under which Automatic Speech Recognition (ASR) is an efficient choice for data entry. In particular the focus was on data entry tasks that are part of constructing military messages. The ADF Formatted Messaging System utilises a structured formatting system to constrain the semantics of a message but also includes a field for unlimited and unstructured text. Hence the data entry tasks involved range from form-filling to free dictation of short phrases. In the experiments, ASR and manual input modes are compared for three data entry tasks: textual phrase entry, selection from a list, and numerical data entry. To effect fair comparisons, the tasks minimised the transaction cycle for each input mode and data type and the main comparisons use only times from correct data entry. The results indicate that for inputting short phrases ASR only competes if the typist's speed is below 45wpm. For selecting an item from a list, ASR offered an advantage only if the list length was greater than 15 items. For entering numerical data, ASR offered no advantage over keypad or mouse. The general conclusion for formatted data entry is that a keyboard/mouse interface designed to match the data to be entered will be more time efficient than any equivalent ASR interface.

Published by

DSTO Electronics and Surveillance Research Laboratory
PO Box 1500
Edinburgh South Australia 5111 Australia

Telephone: (08) 8259 5555
Fax: (08) 8259 6567

AR-012-066
November 2001

APPROVED FOR PUBLIC RELEASE

Experimental Comparisons of Data Entry by Automatic Speech Recognition, Keyboard, and Mouse

Executive Summary

Data entry is the term used to describe the entry of words, numbers or other symbols, into a computer program. Data entry is accomplished via an input mode such as a keyboard, numerical data often being entered via the numerical keypad. Another input mode is automatic speech recognition (ASR) technology, which is available commercially and consequently there is interest in its suitability for Defence use.

The integration of ASR technology with computer programs is occurring within DSTO. This activity motivated the desire to empirically evaluate speech recognition technology for data entry tasks. One such program is Facade, which is speech-enabled ADFORMS software. Although the results are generalisable the data entry requirements of Facade particularly directed the design of this set of experiments. The two questions we attempted to answer are:

1. Whether ASR technology has improved sufficiently for Defence use in data entry tasks, and if not,
2. Investigate whether ASR technology can possibly improve to the extent where it is an efficient replacement for conventional input modes in data entry tasks.

There are three common types of data in ADFORMS: short phrases, numerical data and items in a list. For each of these three data types we compared ASR technology and the currently used input mode. For phrase entry ASR technology was comparable in efficiency only for two-four finger typists. For number entry, ASR technology was not comparable. For selecting an item from a list ASR technology might be comparable if the list was longer than 20 items. These results were found by removing all the data in which errors of one type or another occurred. Consequently they are representative of not only current ASR software but also of foreseeable improvements in ASR technology.

These results do not provide any support to the claim that ASR technology enables more efficient data entry than well-designed conventional interfaces. For tasks with specific requirements, such as a hands-free operator, ASR technology may be useful but this forces consideration of factors such as the impact of ASR errors. Unless a substantial case can be made for the introduction of, and redesign necessary to incorporate ASR technology, we suggest that effort might be better expended by concentrating on optimal but conventional interface design.

Authors

Helen M. Mitchard
Defence Systems Analysis Division

Helen M. Mitchard is a Senior Professional Officer with the DSTO. Her interests include cognitive modelling, applications of artificial intelligence, and human-computer interaction. She has a B.Sc. in cognitive science from Flinders University and a B.CIS(Hons) degree in computer science from the University of South Australia.

Jim Winkles
Information Technology Division

Jim Winkles is a Senior Research Scientist in the Information Technology Division of DSTO, having joined in 1997 from the Centre for Applied Psychology at the University of Canberra. He is a cognitive scientist with an MS in computer science and a PhD in psychology from Stanford University. In 1999 he transferred from the Human Systems Integration Group at Salisbury, South Australia, to the C3 Research Centre in Canberra. Most of his work for DSTO has been in the area of human factors but he is also interested in computational models of cognition, particularly of learning, and in applications of artificial intelligence.

Contents

1. Introduction

1.1 A short summary of research on applications of speech recognition systems

The development of Automatic Speech Recognition (ASR) technology began in the 1950s and commercial systems began to appear in the 1970s. However, compared with pointing devices, especially the mouse, the impact of ASR has been slight. The research to date indicates that ASR will be restricted to "niche" applications where task characteristics dictate that the user's hands and/or eyes are occupied and unavailable for interacting with the computer. More recently, the search has been for multi-modal applications where the availability of an additional response channel increases the efficiency of the interaction by allowing the user to maximise simultaneously the use of both cognitive verbal and spatial resources. Nevertheless, users have always wanted to believe that ASR technology will eventually allow them to talk to their computers. While their enthusiasm is usually dampened by the realities of training a recogniser and correcting recognition errors, optimism persists.

In fact, the evidence that consigned ASR to the margins of interface development was never overwhelming. With the benefit up-to-date ASR software and in the light of lessons learned in twenty years of reported human factors research on ASR applications, the current study revisits some earlier work. The aim is to establish heuristics that specify, at least for routine data entry tasks, the parameters that determine when and how ASR may offer an advantage over other methods of data entry.

1.2 Task characteristics

The motivation for this research is to contribute to an investigation of the suitability of ASR for use with military messaging systems such as the Australian Defence Force Formatted Messaging System (ADFORMS). ADFORMS utilises a structured formatting system to constrain the semantics of a message and thus to decrease the amount of information that has to be transmitted. This structure means that completion of an ADFORMS message is similar to other form-filling data entry tasks.

The information that goes into formatted military messages has a high numerical content and some fields have a limited vocabulary of possible text entries, such as specific place names or the names of items of military equipment, which are of very low frequency in ordinary conversation. However, an ADFORMS message always has one field for unlimited and unstructured text in case material must be included that cannot be conveyed through the formatted body of the message. These characteristics of military messages have important implications for the use of ASR in their preparation.

1.3 Characteristics of speech recognition systems

The following summary should be sufficient for comprehension of the experiments described in this report. A more detailed discussion of the issues outlined here can be found in Simpson, McCauley, Roland, Ruth, and Williges (1985).

The useability of ASR is obviously increased by reducing the error rate of a recogniser. There are two general characteristics of ASR systems that may be used to guide the choice of a particular system for a particular application, speaker dependence/independence and isolated or continuous speech. There is at least one means, restricted vocabulary, by which a recogniser can be configured to improve recognition probabilities using defined sets of words.

A *speaker independent* system is usually only suitable for very small vocabularies such as the simple command set of a videocassette recorder. By contrast, a *speaker dependent* system is adapted to the speech patterns of individual users. Such a system maintains a profile of each user that contains a record of any special features of the user's speech which are relevant to the word recognition algorithm. These features may include representations of non-standard pronunciations of certain phonemes and of whole words that have required special training before they were recognised. The profile is adapted by an *enrolment* procedure when the user first encounters the system and is updated automatically when misrecognitions are corrected. Enrolment consists of training a set of words and phrases specified by the manufacturer of the recogniser. A speaker dependent system was chosen for the experiments because military messages can involve a large vocabulary.

An *isolated word* system stores representations of individual words as strings of phonemes to be compared with the phoneme strings inferred from a user's utterances. A *continuous speech* system uses word-word transition probabilities to improve the recognition of each word by virtue of its neighbours. (In a sense, the recognition unit of a continuous speech system is a phrase rather than a word.) Both systems may use some form of syntax checking to narrow the possibilities for a particular word in a sentence but this may be more useful to a continuous speech recogniser because it has succeeding as well as preceding words from which to work out the syntax of the utterance. An isolated word system was chosen for the experiments because the components of military messages are often cryptic rather than consisting of the properly formed sentences for which continuous speech systems are optimised.

If, in a particular application, the complete set of words to be recognised is known in advance, a vocabulary can be created using that set. The recogniser can be trained on just that vocabulary and using this vocabulary will increase recognition probabilities. In *command mode*, the recogniser is configured to recognise every utterance only as one of such a pre-trained *command set* or not all. Command mode gets its name because it is often used as an alternative to mouse activated menu commands but it can also be

used, as it was in the experiments, for choosing a form entry from a limited number of alternatives.

With the recogniser used in the experiments, the alternative to command mode is called *dictate mode*. In this mode the recogniser has available a very large vocabulary of word templates. In addition it may offer a list of alternative words in addition to the word nominated as that most likely to have been (the one) spoken. This allows the user to select the correct alternative (if listed) rather than repeating the word to be recognised. One of the experiments tested the efficacy of dictate mode as an alternative to typing.

A facility for training defined sub-vocabularies exists in dictate mode in which not every word has to be trained. (For instance, if the sub-vocabulary is the names of the integers from one to one hundred, it may only be necessary to train on the digits and the tens in order to increase the recognition probabilities of the whole range.) In use, alternatives may be offered from the complete vocabulary, although the sub-vocabulary will be favoured because of the training. This facility was employed in one of the experiments.

1.4 Research objectives

There are two characteristics of any mode of data entry for a form-filling task which are crucial to determining its efficacy. They are the speed of data entry and the error rate associated with that process. At first glance the speed of human speech would suggest that ASR technology must revolutionise data entry. An initial experience with ASR will quickly reveal that error rates can be very high and that error correction procedures can be clumsy and time-consuming. A little more experience, however, shows that error rates and error correction delays can drop dramatically as both the user and the recogniser adapt to each other.

Thus, two questions must be answered to give a complete picture of the usefulness of ASR. They are (1) How quickly do error rates decline to an acceptable level? and (2) What rates of data entry can be obtained when error rates are acceptably low? If the answer to the second question reveals that the best rates of data entry by ASR are slower than those obtainable by other means then ASR will be limited to applications for which rapid data entry is not a primary consideration and the first question is worth considering only in those contexts. Consequently, the experiments reported here were designed specifically to answer the second question. That is, the dependent variable in each experiment is the time to enter an item which does not require correction. Using timing data from uncorrected items in these experiments to represent the best times obtainable with ASR depends on an assumption that the effects of the mutual adaptation of user and recogniser are limited to error rates and error correction. This assumption was supported by pilot work and further strong support is provided in section 3 by data from the Phrases Experiment.

Of course, as a byproduct of studying correct data entry a good deal of data was obtained on the time to enter corrected items. Although only the early stages of mutual adaptation by user and recogniser were observed, this data has some bearing on the questions of how steep are the declines in error rates and in delays due to error correction. Some suggestions for a different set of experiments to address these questions more completely are contained in section 6 below.

2. General design and conduct of the experiments

2.1 The perils of evaluation studies

The initial inquiry that motivates an evaluation study can be deceptively simple. (For instance, the research reported here arose out of a request along the lines of "Find out if ASR could be useful for military messaging systems".) In order to address the general question through experimental studies it must be operationalised into test tasks and measuring instruments to be employed with a particular set of users. This creates a gulf between the vague, common sense notion of what is to be tested and its realisation as a precise set of experimental questions. As a result the claims that are eventually made about the conclusions from an evaluation study by the commissioners of the study, or by casual readers, are often far more general than can be supported by the data.

Evaluation researchers are usually fully aware of limitations in their studies which are forced on them by restrictions of available resources or representative participants but these caveats are not always repeated when their results are reported by others. In addition, when research on a topic proceeds over many years, concepts are developed which can, in hindsight, make earlier work appear naive or misguided. A substantial literature on ASR applications has grown up over the last thirty years. However, very little of it bears directly on the issue of data entry speeds and (in the light of 20/20 hindsight) some of the experimental comparisons are flawed by the use of atypical participants or by differences in task requirements when ASR is compared with an alternative. No study of which we are aware has examined separately the differing demands of entry of different data types nor are we aware of any experimental comparisons that have employed commercial ASR software of the speed and quality now available. We mention below some studies and critiques that have influenced the design of our experiments. A more detailed review can be found in Damper and Wood (1995).

Welch (1977) concluded that for simple data entry tasks, keyboard provided a faster input mode than ASR. The experiment was designed to compare aspects of data entry such as speed, accuracy and correction times. The input modes were keyboard, ASR

and graf pen, a pointing device. The data entry task was classed as simple (copying) or complex (requiring parsing by the participant). Our focus is on the simple data entry task. The skill distribution of participants in the simple data entry experiment, mostly highly experienced or expert typists, gave the keyboard "a distinct advantage" (Welch, 1977, p17). Another advantage conferred on the keyboard condition arose from a reduction in the recommended ASR training.

Similar results to those of Welch (1977) were reported by McSorley (1981, cited in Simpson et al, 1985) who found manual data entry to be faster than using ASR. For tasks that are more complex, the situation is less clear cut. Leggett and Williams (1984) evaluated a restricted vocabulary ASR as an input mode for input and editing tasks. For both types of tasks participants completed more tasks, input and editing, using keyboard as opposed to ASR, however the use of ASR resulted in fewer errors for both tasks. They concluded that given more experience with ASR it could be a competitive input mode.

Morrison, Green, Shaw, and Payne (1984) compared ASR and keyboard performance times for a text-editing task. They found no significant differences in task times between the input modalities or the editors. The text editors differed in their transaction cycles, defined as the number of commands required before a system responds. An 'almost significant' result indicated that for ASR, short transaction cycles might be optimal.

In a simulated naval command and control task, Poock (1980, 1982, cited in Damper & Wood, 1995) reported a highly significant advantage for ASR over typing, both in terms of input speeds and error rates. However, Damper and Wood (1995) conducted similar experiments in which they claimed a fairer comparison by using abbreviated commands in the typing condition in order to minimise the transaction cycle. In contrast to Poock, they found that speech had significantly more errors than keying but there was no significant difference in input times.

2.2 Matching entry modes to data types

The entry of three data types comprise a large part of completing an ADFORM and are found in many similar form-filling data entry tasks. They are lists, phrases, and numbers. A list data type means a (defined and) limited set of possible entries, often proper nouns such as place names or the names of pieces of equipment. Phrases means strings of words that form (possibly cryptic) phrases or sentences. Numbers may include decimal points, such as latitudes and longitudes, and may be signed, as in positive and negative temperatures.

For each of these data types there are "traditional" data entry tools. For lists, there is selection by mouse (or possibly by arrow keys). For phrases, there is the standard QWERTY keyboard and for numbers, there are the digits at the top of the standard

keyboard or in the numeric keypad on the right hand side of the extended keyboard. A number of experimental conditions were constructed to compare data entry by ASR with more traditional methods for each of the three data types. The conditions are summarised in table 1.

Table 1. The experimental conditions.

Data type	Modes of data entry	ASR vocabulary trained in advance
Lists	Mouse selection	
	Command mode ASR	All items in all lists
Numbers	Numerical keypad	
	Mousepad	
	Command mode ASR	Digits 0-9 and required special characters
	Dictate mode ASR	Double digits 0-99 and required specials
Phrases	Keyboard	
	Dictate mode ASR	(No pre-trained vocabulary)

Command mode ASR was chosen for use where the required vocabulary was small enough to be exhaustively trained before any data was entered. For the double digits condition, the recogniser used in dictate mode has access to the user information gained while training double digit numbers in a numerical sub-vocabulary. The appropriate numerical sub-vocabulary was chosen from those available in dictate mode on the basis that the numbers contained reflected the manner by which data was input. The recogniser usually required user training only for the numbers 0-23, 30-33, 40-43, 50-53, 60-63, 70-73 and 80-83.

2.3 Experimental design considerations

Each experimental task consisted of a number of data items to be copied one at a time from one part of the interface to another part immediately below using the input mode for that condition. Several blocks of items were entered in each speech condition in order to examine the early effects of adaptation by the recogniser to the user's voice (and the adaptation of the user to the recogniser). Only one block of items was used for the "traditional" mouse and keyboard entry modes in the Lists and Phrases Experiments respectively because little learning was expected in those. In the Numbers Experiment, however, all entry modes had some degree of novelty and two blocks of items were entered in each of the four conditions.

In order that no extraneous differences were introduced due to variations in eye movements or memory requirements, the interface for each experiment had the same appearance for each condition. Damper and Wood (1995) pointed out that in some

experimental comparisons between ASR and other input modes the experimental tasks had employed input which was optimised for one mode but not another. They recommended minimisation of the transaction cycle (the sequence of operations that have to be performed to complete the task) for each mode. That principle was employed in the experiments reported here.

Participants in pilot experiments had some difficulty adapting to the use of the speech recogniser. Their difficulties were exacerbated if the error rate of the recogniser was high so that they were continually involved in correction and retraining. However, if they first experienced ASR in command mode with a small vocabulary, the lower occurrence of errors reduced their apprehension. This increase in confidence led to a more natural manner of speaking which increased consistency, further reducing errors.

For this reason, to maximise the number of correctly completed tasks and to limit participant distress in an extended experimental session, the experiments were ordered by increasing vocabulary size, first the Lists Experiment, then the Numbers Experiment, and finally the Phrases Experiment. However, each participant was enrolled as a new user, with a new user profile, for each ASR condition. This ensured that the data collected in each condition was independent of recogniser training in earlier conditions.

All participants completed all conditions in the experiments and the order of conditions was counterbalanced within each experiment. As mentioned above the two ASR conditions were ordered by vocabulary size and for convenience used in succession. However, the other input modes were counterbalanced with the ASR grouping within each vocabulary size. After completing all of the conditions, each participant also completed a test of typing speed, which was used to calibrate their typing behaviour in the Phrases Experiment.

2.4 Equipment

The computer was a 333MHz Pentium II with 256 MB of RAM and a SoundBlaster Creative Labs PCI-128 sound card. The speech recognition software was Dragon Dictate's Classic Edition 2.5. It is an isolated word, speaker dependent recogniser that includes a noise-cancelling microphone. In dictate mode Dragon Dictate has 30,000 words in the active vocabulary and 120,000 words in the backup vocabulary.

The enrolment procedure for Dragon Dictate creates a new user profile by setting microphone levels and training 16 preset words and phrases. The enrolment process takes approximately 3 minutes. The training set is reproduced in Appendix D.

The *latency* of the recogniser is the time taken to register that an utterance has been made and to display the most probable word. This is obviously a function of both the hardware and the software used and possibly also of the vocabulary and the speaker.

Simple timing trials (repeating a short word several times with and without the recogniser) indicate that the latency of the system described is at most 400ms.

The latency issue is not as simple as it appears because, in isolated word recognition, time is also consumed by the human cognitive processes which occur as the speaker processes the appearance of the displayed word and prepares for the next utterance. In ordinary use, the total delay between the completion of one utterance and the start of the next is typically 600-800ms, only part of which is taken up by recogniser latency. Data from the Numbers Experiment examined in Section 6.3 shows that the total time to speak and process a monosyllabic word is around 1200ms. The time to utter such a word in the rather deliberate way that one speaks to a recogniser is around 300-500ms and recogniser latency is at most 400ms. The balance is taken up with cognitive processes such as reaction to the displayed word and retrieval of the next word to be uttered, either from short term memory or by saccadic eye movement to the stimulus display.

2.5 Participants

In response to an email that made no mention of ASR, 24 people volunteered to be participants in a data entry experiment. All participants were psychology or computing professionals from within DSTO with no previous ASR experience. In total, there were 5 females and 19 males over an age range of 18-45 years.

2.6 Procedure

The experiments were conducted with one participant at a time in a quiet office environment. Participants were advised that the purpose of the experiments was to compare data entry speeds for a form filling task using different modes of data entry, with an emphasis on comparing ASR with more traditional modes. They were told that they would be asked to enter several sets of items as quickly as possible using a variety of modes and that they were free to take breaks as necessary between items and between sets of items because only the time to complete each task was recorded by the computer.

The particulars of training, interface and conditions for each experiment are described in Sections 3, 4, and 5 below. The following procedure was employed for every condition.

- If it was a speech condition, the participant was enrolled as a new user and the appropriate ASR training was completed.
- The experimenter activated the appropriate program and the appropriate file containing the input was loaded.

- A demonstration task was initiated by the experimenter who then paused to describe the input display, the data entry interface and the manner in which the task was to be completed.
- The experimenter completed the demonstration task and two more error free tasks.
- The experimenter described and demonstrated the error correction procedures for that condition using two more tasks.
- The participant was instructed to complete the first set of tasks.
- After the first few items the participant was reminded and encouraged to complete the tasks as quickly as possible.
- Further sets of tasks were completed by the participant, as required, with repetition by the experimenter of the instruction to enter the data as quickly as possible.

2.7 Automatic data recording and analysis

The Dragon Dictate software includes Xtools 2.0, a program development environment which enables interfacing between Dragon Dictate, Visual Basic and Delphi C++. The programs for presenting the experimental tasks and collecting data were written in Visual Basic. Keyboard, mouse and ASR events were logged to nearest 10ms using a special routine written in Delphi C++. A dedicated machine was used to ensure a constant response time by ASR and all event data was downloaded to log files for later analysis. Speech events generated by DragonDictate are converted into keyboard events and so the log provides a uniform record of data entry in the speech, keyboard, and mouse conditions. Errors and corrections in data entry were determined from the log together with the time for each participant to enter each datum in each condition.

While automatic logging provides a rich and reliable data file, it does have the disadvantage that pauses and hesitations during a data entry task are recorded along with uninterrupted efforts. If a participant breaks off in the middle of a task to ask a question of the experimenter or to take a drink of water, the time for the interruption cannot be excluded from the log. In fact, most such interruptions occurred between tasks rather than during a task but the log still shows exceptionally long times for some tasks.

Means and standard deviations were computed for correct and corrected items in each condition (and in each field of each condition in the Numbers experiment) in order to identify times lying more than three standard deviations from the mean. Appendix B summarises the number and distribution of these outliers. Outlier times were examined first to eliminate the possibility of errors during data reduction from the automatic log to the spreadsheets of times for each participant on each item. They were then examined for patterns attributable to particular participants but none were found to have contributed more than 5 outliers in the Lists Experiment, the Phrases Experiment or the totals in the Numbers Experiment, and the highest contributors in each experiment were not high contributors in the others. In fact, there were fewer than 30 outliers in each experiment, more than half of the 24 participants contributed

outliers in each experiment and only one participant contributed no outliers to any of the experiments. Finally, checks were made to see if particular items had given rise to large numbers of outliers. There was a predictable tendency for the first item in each new input modality to give rise to more outlier times than other items. Other than that, no patterns were found.

It was decided that it would be unwise to allow all of the outlier times to be used in the data analyses because some of them are very extreme. (See Appendix B.) Since the majority of them were in speech conditions, this might also have the effect of biasing the analyses against speech entry. Given the large number of data points, dropping outliers more than four standard deviations from the mean of each condition was chosen as a strategy to exclude the obvious anomalies without unnecessarily truncating the spread of the data. Using this strategy, only 1% of data points were dropped.

3. The Phrases Experiment

This was the last of the experiments to be completed by participants. It is reported first because its results have implications for understanding the results of the other two experiments.

3.1 Interface, stimuli and experimental conditions

In this experiment, the two conditions were the input modes of keyboard and ASR. Half of the participants completed the keyboard condition first and half completed the ASR condition first. The interface for the ASR condition of the experiment is shown in Figure 1. The interface for the keyboard condition differed only by the absence of the choice list provided by the recogniser. The experimental protocol is described in Section 2.6.

The stimuli were short phrases of 2-5 words such as "RETURN COURSE TO SQUADRON", "TARGET CHANGE" or "APPROACHING FIGHTER". A complete listing of the phrases used can be found in Appendix A. As the participants all had reasonable keyboard experience, they completed only one block of twenty keyboard items. In the ASR condition, participants completed four blocks of twenty items. (Pilot testing had established that at least three blocks of items were necessary for the error rate to reach an asymptote.)

Immediately after the keyboard block, each participant completed a minimum of three typing tasks at an accuracy of greater than 95%, to determine his or her typing speed. The typing test was Broderbund Type! that was installed on a Macintosh PowerBook 170 with an extended standard keyboard attached.

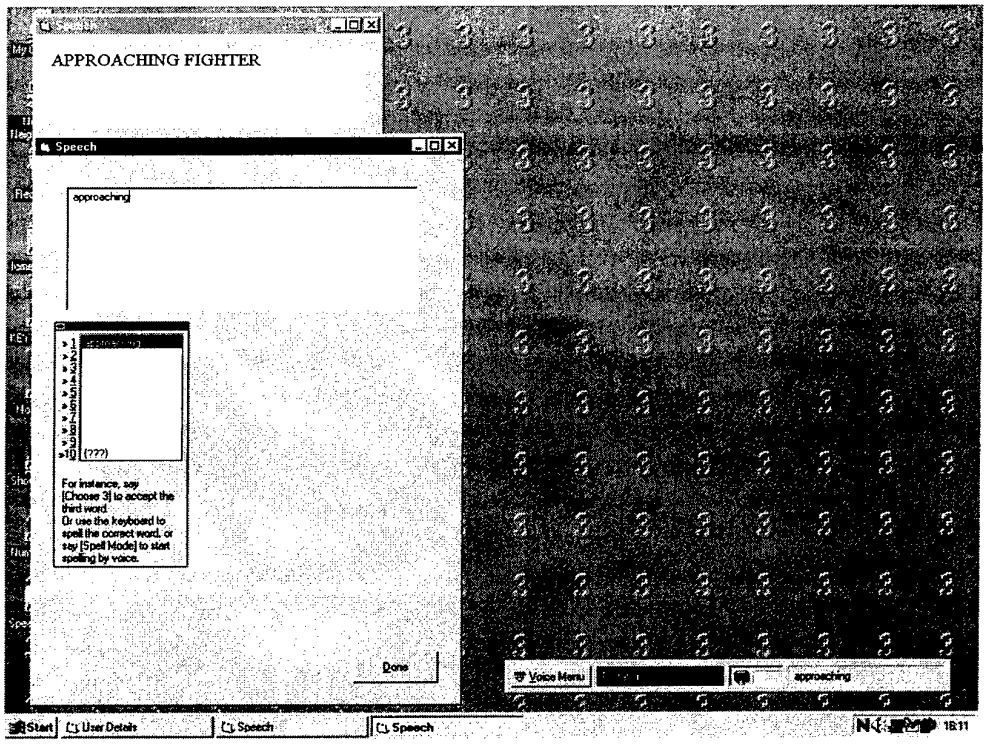


Figure 1: Graphical user interface for Automatic Speech Recognition condition of Phrase Experiment.

3.2 Data entry and correction

In the keyboard condition, participants simply typed each phrase into the response window, pressing the "ctrl" and "d" keys simultaneously to signify the end of the phrase. If an error was detected before the phrase was completed it was corrected using the backspace key and the left and right arrow keys as necessary. Errors detected after pressing the "ctrl" and "d" keys could not be corrected. The next phrase appeared in the stimulus window when the participant pressed the Enter key.

In the ASR (dictate mode) condition, the words in the phrase were entered one at a time with slight pauses between words due to the use of an isolated word recogniser. At the end of the phrase the participant said "done". The next phrase appeared in the stimulus window when the participant pressed the Enter key.

Each time that the recogniser registered that an utterance had been made, a choice list like that in Figure 1 appeared listing, in decreasing order of probability, the possibilities for the word that might have been spoken. The word at the top of that list (if any) also appeared in the response window. If the word in the response window was not correct but the correct word was available in the choice list, the participant could choose the correct word by saying (for instance) "choose three" and that choice

would be inserted in the response window. If the correct word was not available in the choice list, the participant could remove the incorrect word (if any) from the response window by saying "scratch that" and continue the task by repeating the correct word. If, after three attempts, the recogniser did not register the correct word by this procedure the word was immediately trained by that user using *in-line training*.

The experimenter controlled in-line training of a word. The participant ceased attempting to enter data, the required word was found in the dictate mode dictionary (or typed in if it was not already present) and the participant spoke the word three to five times on cues from the recogniser. Then the participant recommenced the attempt to enter the word into the response window. In-line training was usually completed in about twenty seconds, time that was included in the data log.

3.3 Results

3.3.1 Typing performance

Participants' speeds on the typing test ranged from 22 to 70 wpm. Just over half were in the range of 20-40 wpm.

The situation in the keyboard condition of the experiment was slightly different from that in the typing test. During the test, participants had to copy text from the screen so that those who were not touch typists had to switch visual attention back and forth between the screen and the keyboard. By contrast, the phrases used in the experiment were simple enough to be held in short term memory so that attention switching was not required. It is, therefore, necessary to establish whether typing performance in the experiment was similar to ordinary performance as exemplified in the test.

Typing speeds in the typing test were all recorded at above 95% accuracy and so they can be compared with participants' performance on phrases typed without corrections in the experiment. Since the phrases are too short to allow meaningful typing speeds to be calculated in words per minute, the number of characters were counted in each phrase and a speed was calculated in characters per second. The correlation between typing speeds from the typing test (in wpm) with typing speeds from correctly entered phrases (in chars/sec) was 0.94. That is a sufficiently high to conclude that typing performance in the experiment was very close to "natural" performance and that conclusions from the experiment that relate to typing speed are transferable beyond the conditions of the experiment.

Participants in the experiment were instructed to enter data as quickly as possible. Of course, this led to rather more typing errors than might otherwise have been the case but also allowed a meaningful comparison of error correction performance between the experimental conditions. Seventy-two percent of the phrases were typed in correctly without corrections and a further 24% were entered correctly with some correction. The

correlation between typing speeds in the experiment and the number of phrases typed incorrectly by each participant was about 0.4, indicating a slight tendency of faster typists to trade speed for accuracy.

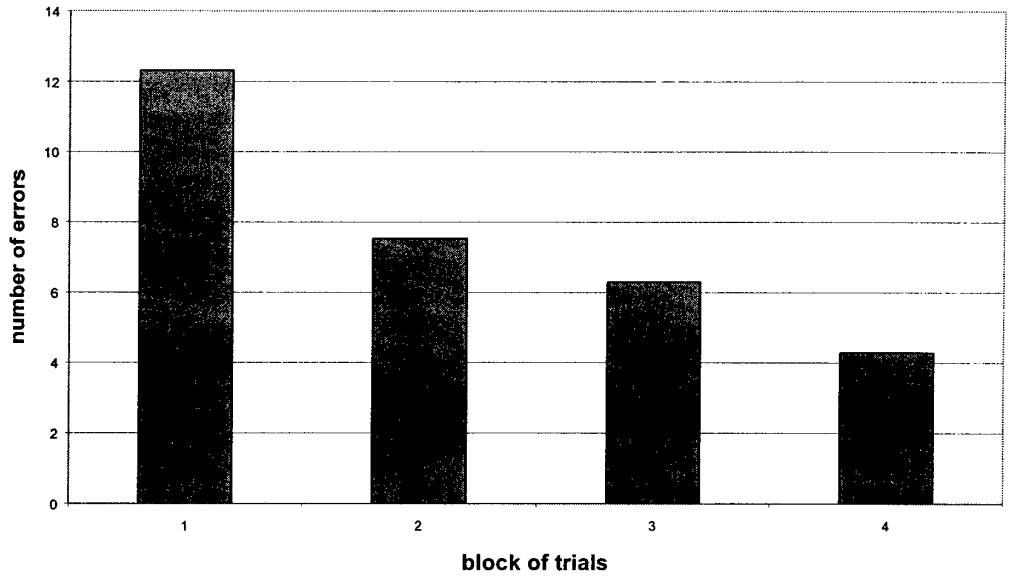


Figure 2: Reduction in errors over four blocks of trials in the Automatic Speech Recognition condition of the Phrases Experiment.

3.3.2 Speech performance

Data entry using ASR was novel to virtually every participant. Looked at from the other direction, each participant was a novelty for the speech recogniser. Each became adapted to the other over the course of the experiment. The most obvious sign of the novelty was a very high rate of recognition errors on phrases in the first speech block. The most obvious sign of the adaptation was the dramatic reduction in the rate of recognition errors in the succeeding blocks. Those effects are illustrated in the column graphs of Figure 2.

The reduction in the average time to enter a phrase across the four speech blocks is illustrated in the column graphs of Figure 3 which includes times for all items entered correctly first time or after correction. The source of the obvious learning effect is brought out much more clearly in Figure 4 in which times for correct items are compared with times for corrected items. The time to enter a phrase correctly falls only very slightly over the four blocks. (This supports an assumption underlying all of the experiments in this report, namely that performance on correct items in the

experiments is representative of data entry using ASR and not simply representative of the performance of ASR novices.) By contrast, the time to enter a phrase that must be corrected almost halves.

Although some simple comparisons will be made, a proper evaluation of error rates and data entry times by ASR which involve error correction would require a longitudinal study, some possibilities for which are discussed in Section 6.3. However, there is a fundamental difficulty in reducing error correction time in ASR. The unit of entry in ASR is a word. Correction of a recognition error requires at least a command to delete the erroneous entry ("scratch that") and repeating the word, guaranteeing that the total time to complete entry of a corrected word is at least three times the time for a correct entry. If retraining of the word is required an additional 20s or so is needed. Efficient data entry by ASR is thus dependent either on reducing errors to a negligible rate or on other (non-speech) techniques for error correction.

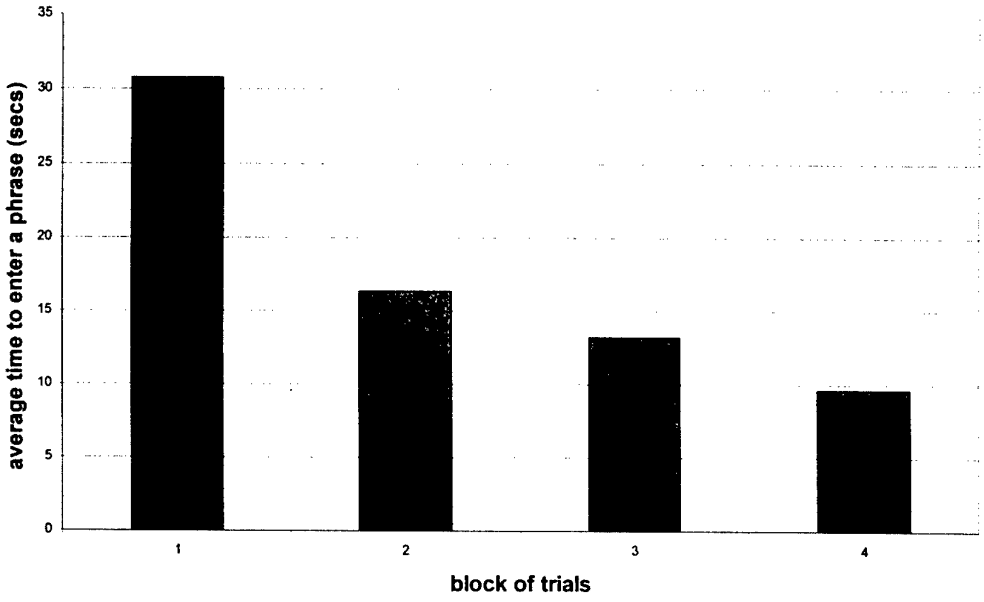


Figure 3: Average entry times for a single phrase over four blocks of trials in the Automatic Speech Recognition condition of the Phrases Experiment (includes times of both correctly entered and corrected phrases).

Figure 4 suggests a parallel between text entry using ASR and text entry using a manual typewriter. Like ASR entry errors, most ordinary typing errors are noticed immediately by the typist but also like ASR, error correction on a manual typewriter is very time consuming. The principal advantage of an electric typewriter with (at least) one line of memory is not greater speed in the entry of correct text but greater speed of error correction when the error is detected before it has been committed to paper. This

advantage is, of course, retained in full screen word processors and word processing software on computers where the additional cut and paste facilities assist more in composition than in error correction.

Typists trained exclusively on modern machines would have to adjust their technique if required to use manual typewriters. In particular, they would probably slow down in order to minimise errors and costly corrections. The same sort of effect is observable when ASR users adjust their speaking patterns to the recogniser. A longitudinal study could determine the amount and type of practice and adjustment necessary to reduce error rates using ASR to very low levels at which the high cost of correcting an occasional error would not be important. Such qualitative and quantitative information about training requirements could be used to assess the suitability of ASR for particular applications.

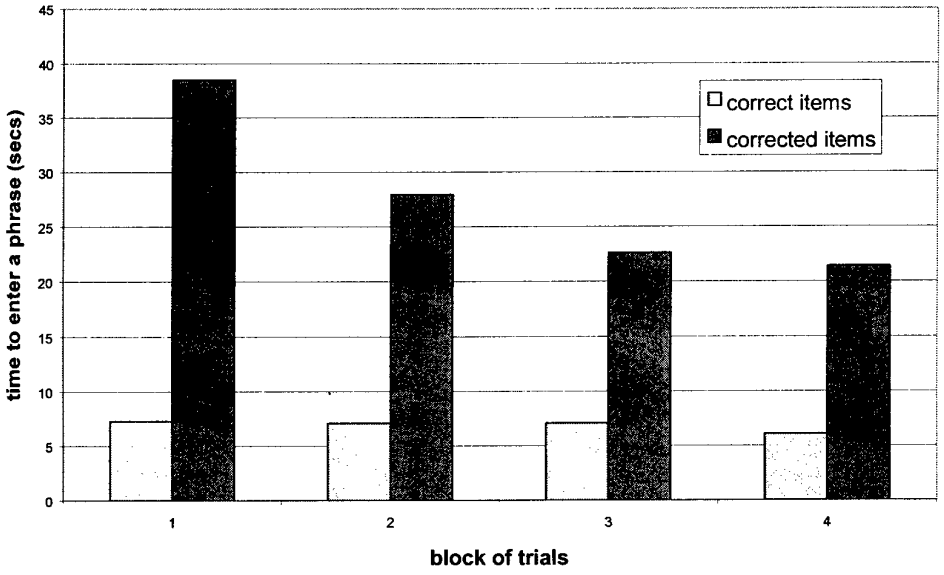


Figure 4: Average entry times for a single phrase, without or with corrections, over four blocks of trials in the Automatic Speech Recognition condition of the Phrases Experiment.

3.3.3 Comparing speech performance with typing performance

Seventy-six percent of phrases were entered correctly without correction in the fourth speech block compared with 72% in the keyboard condition. However, the comparison is somewhat misleading because the unit of entry (and hence of error) is a word in the speech condition but a letter in the typing condition. For instance, the phrase "NEW COURSE REQUIRED" presents only four opportunities for error by the recogniser

(including the instruction "done" at the end of the phrase) but 21 opportunities for the typist to err (including the final "ctrl" and "d" keystrokes).

In Figure 5, bars have been added for correct and corrected entries in the keyboard condition to those for the speech blocks shown in Figure 4. It illustrates that if errors are eliminated, speech entry can be as quick as or quicker than typing but it also shows how errors consume time and increase user frustration. A particular difficulty with error correction by speech is that it consumes verbal resources in working memory (Karl, Petty & Schneiderman, 1993) in a way that error correction by typing does not, except in the most inexperienced typists. This can disrupt cognitive processing of even the simplest messages.

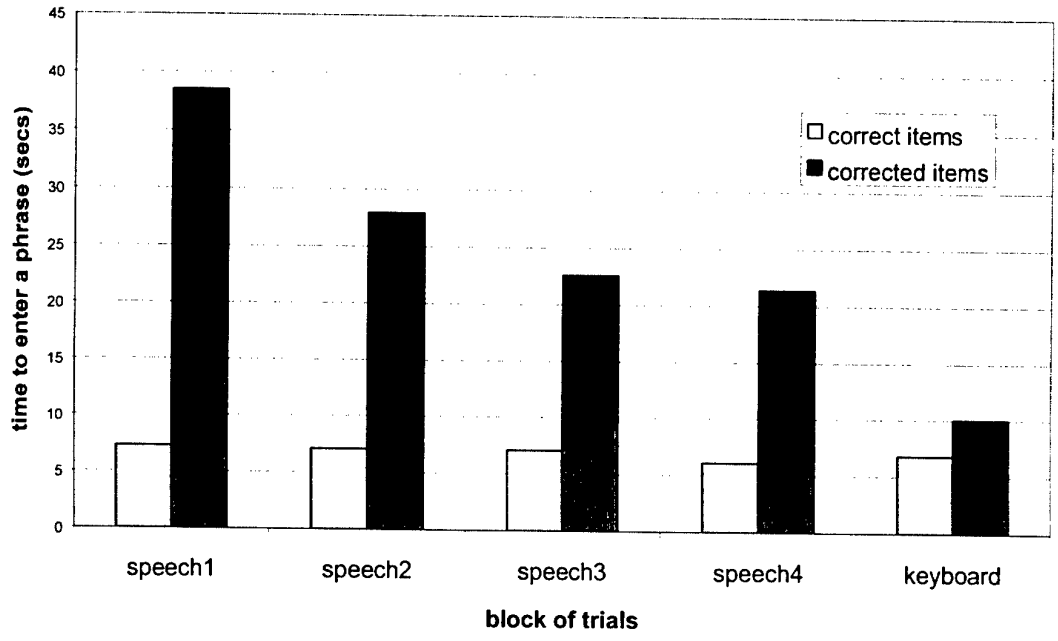


Figure 5: Average entry times for a single phrase, without or with corrections, over four blocks of trials in the Automatic Speech Recognition condition of the Phrases Experiment and one block of trials in the Keyboard condition.

A more revealing comparison of entry speeds for items entered correctly first time is shown in Figure 6 in which each participant's average entry time from the typing condition and from the last speech block are plotted against typing speed. As would be expected, entry time by ASR is independent of typing speed (correlation = -0.28, $p<0.1$) but entry time by keyboard is a linear function of typing speed (correlation = -0.90, $p<0.0005$). The crossing of the fitted linear regression graphs suggests a rule of thumb that if you type faster than 45wpm you're quicker to do that than even error-free speech entry. If you're a slower typist, you would be quicker by speech entry if you

could do it without errors. (Since the fit of the linear regression line to the speech times is weak, this conclusion warrants checking in more detail. In fact, only four participants violated the rule. All had typing speeds close to 45wpm and in three cases, the difference in times was less than 10%.) Such a heuristic has obvious implications for staff selection and assignment when ASR is under consideration for any data entry application or for choice of data entry technology where staffing is already determined by other considerations.

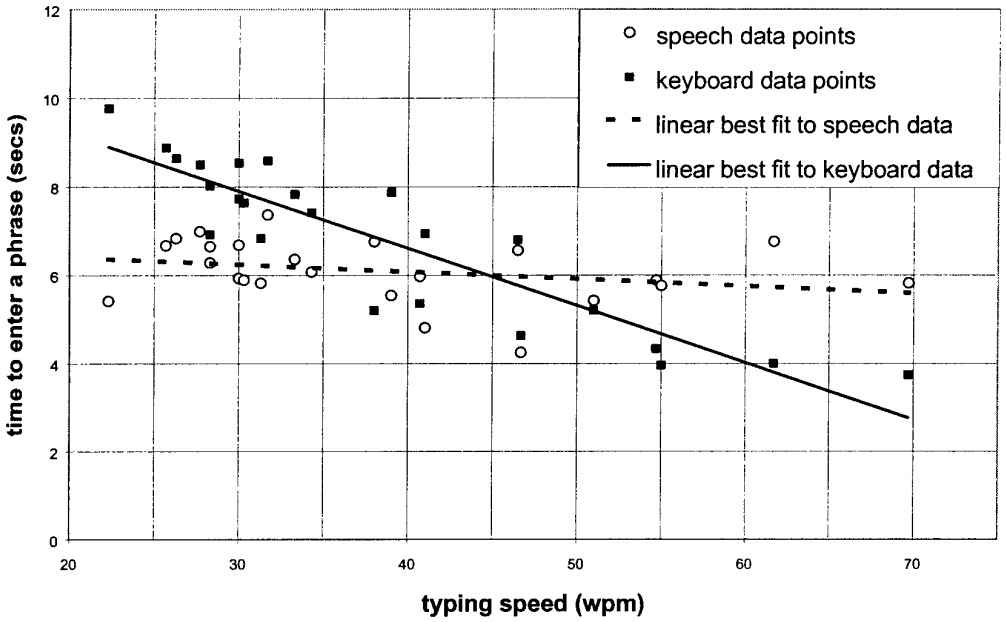


Figure 6: Average times for entry of a single phrase without corrections in the Keyboard condition of the Phrases Experiment plotted against participants' typing speeds and compared with their average times to enter a phrase without corrections in the fourth block of trials in the Automatic Speech Recognition condition also plotted against typing speed.

4. The Lists Experiment

4.1 Interface, stimuli and experimental conditions

In this experiment, the two input modes were mouse and command mode ASR. Half of the participants completed the mouse conditions first and half completed the ASR conditions first. The interface for both input modes, with the condition using 25 item lists, is shown in Figure 7. The task is to input the indicated triplet consisting of a city of origin, a destination city and a type of aircraft by selecting the correct items from the three lists. The lists of cities of origin and destination are the same. Twenty triplets

were entered in each of the five conditions. The protocol by which the experiment was conducted is described in Section 2.6.

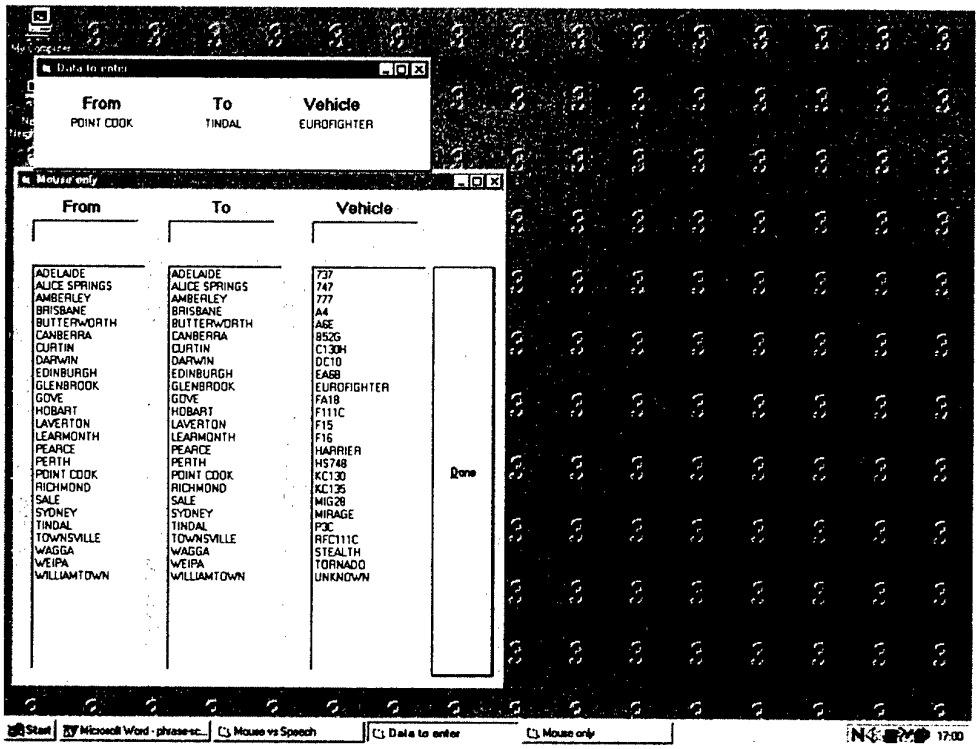


Figure 7: The interface for the Automatic Speech Recognition condition of the Lists Experiment.

In the mouse conditions participants completed tasks using short lists (5 items), medium lists (10 items) and long lists (25 items). The short and medium lists are shown in Table 2 below. As time to speak an item does not vary with list length, only the short and long lists were used as ASR conditions. The conditions were ordered by list length to optimise participants' experience with ASR. As the available vocabulary for each ASR condition was equivalent to the list length, the smaller vocabularies offered a higher probability of recognition.

4.2 ASR training

For the List Experiment, ASR training consisted of training a vocabulary of 53 words that was used in the long list condition (25 cities, 25 aircraft types and three commands, "next", "back" and "done"). This process also trained the short condition in which the lists are subsets of those in the long list condition. The command mode training consisted of repeating each item 3-5 times.

Table 2: Short and medium lists for the Lists Experiment.

	Cities	Aircraft types
Short Lists	ADELAIDE	B52G
	CURTIN	DC10
	DARWIN	EUROFIGHTER
	PEARCE	FA18
	TINDAL	MIRAGE
Medium Lists	AMBERLEY	737
	CANBERRA	A6E
	EDINBURGH	C130H
	GOVE	DC10
	LAVERTON	FA18
	PEARCE	HARRIER
	RICHMOND	MIG28
	SALE	P3C
	TINDAL	RFC111C
	WILLIAMTOWN	UNKNOWN

4.3 Data entry and correction

In the mouse conditions, participants clicked on the three items of a triplet in the appropriate columns and then clicked on the "Done" button. If an error was detected at any time before the "Done" button was clicked it could be corrected immediately by clicking on the correct item in the appropriate list. That is, consistent with minimisation of the transaction cycle for mouse selection, the focus in the data entry fields was determined by the column in which a mouse click was detected. A new triplet was displayed when the participant clicked on the "start" button.

In the ASR conditions, participants spoke the three items of a triplet in order, with slight pauses between words for the recogniser to respond, and then said "done". That is, consistent with minimisation of the transaction cycle for correct entry, the focus in the data entry fields moved from left to right as words were recognised. If an error was detected at any time before "done" had been said it could be corrected immediately by transferring focus back to the appropriate field by saying "back" once or twice, saying the correct word, and if necessary moving the focus back to the unfilled field by saying "next". A new triplet was displayed when the participant pressed the return key.

Occasionally the recogniser did not register the correct word after three attempts, so the word was specifically trained for that user by means of *in-line training*, very similar to that used in the Phrases Experiment. The experimenter controlled in-line training of a word. The participant ceased attempting to enter data, the required word was found

in the command mode dictionary and the participant spoke the word three to five times on cues from the recogniser. Then the participant recommenced the attempt to enter the word into the response window. In-line training was usually completed in about twenty seconds, time which was included in the data log.

4.4 Comparing speech performance with mouse selection performance

The average number of errors made by participants in entering each set of twenty triplets was almost six per set in the two speech conditions (large list 5.92, small list 5.71) compared with about one per set in the three mouse selection conditions (large list 1.38, medium list 0.88, small list 0.67). However, the comparison between error rates in the two modalities is misleading because of the small amount of recogniser training employed. Figure 2 above indicates the sort of rapid reduction in error rates that could be expected with further recogniser training and with the limited recognition vocabulary of this experiment, recognition errors could quickly be reduced to very low rates.

The comparison of interest in this experiment is between the times to correctly enter a triplet in each condition. The relevant data are illustrated graphically in Figure 8. Entry time in the mouse selection conditions is clearly a linear function of list length whereas entry time by ASR is independent of list length. Both results are predictable from elementary considerations of ergonomics and the six seconds or so required to enter triplets of three words by ASR in this experiment corresponds closely to the time to enter phrases of similar length in the Phrases Experiment (see Figure 5 above). The main interest lies in the fact that the graphs cross verified by one-tailed comparisons of the means for each input modality, $t(23)=3.15$, $p<0.0025$ at the upper end, $t(23)=5.26$, $p<0.0000125$ at the lower end), suggesting another rule of thumb that selection from a list shorter than about 15 items is quicker by mouse for most people whereas selection from a longer list is quicker by ASR.

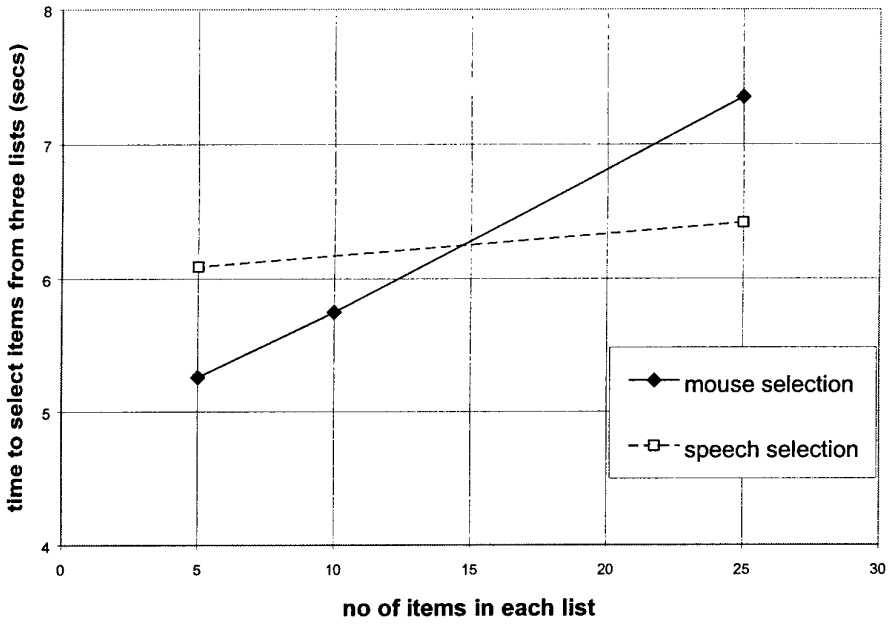


Figure 8: Average time to select a triplet of words (without errors) plotted against list length for the three Mouse Selection conditions and the two Automatic Speech Recognition conditions of the Lists Experiment.

5. The Numbers Experiment

5.1 Interface, stimuli and experimental conditions

In this experiment the input modes were keyboard, mousepad, command mode ASR and dictate mode ASR. As explained above, the command mode condition always preceded the dictate mode condition because it helped participants to become familiar with ASR if they used it first with a smaller vocabulary. The grouping of the two ASR conditions was counterbalanced across participants with the other two conditions. The protocol by which the experiment was conducted is described in Section 2.6.

Numeric data entry using the keyboard is available in two forms, the numeric keys at the top of the QWERTY keyboard and the numeric keypad on the right hand side of the extended keyboard. Despite a lack of experience with the keypad and the need to use three punctuation keys, pilot testing established that participants entered numeric data more quickly using the keypad rather than the keys at the top of the keyboard. Consequently, the keypad was adopted for the keyboard input condition of this experiment.

The mousepad was a mouse-activated on-screen equivalent of the keypad. It had the same layout and dimensions as the keypad with the addition of buttons for the necessary punctuation and navigation commands and resembled a software calculator.

In the command mode condition, participants input numeric data using single digits. A dictate mode ASR condition was also implemented in this experiment for several reasons. Firstly, it enabled participants to enter data in a more natural manner than digit by digit. Secondly, it is an approximation of continuous speech ASR. Thirdly, in dictate mode, participants used the recogniser's 30,000 word vocabulary, although admittedly only a small subset. This enabled some assessment of the effects of vocabulary size in comparison with the small vocabulary used in the command mode condition. Although dictate mode allows multi-digit entry, in order to keep errors to an acceptable level, participants were asked to use double-digit entries (combined with single digits for numbers with an odd number of digits). This decision was based on pilot work.

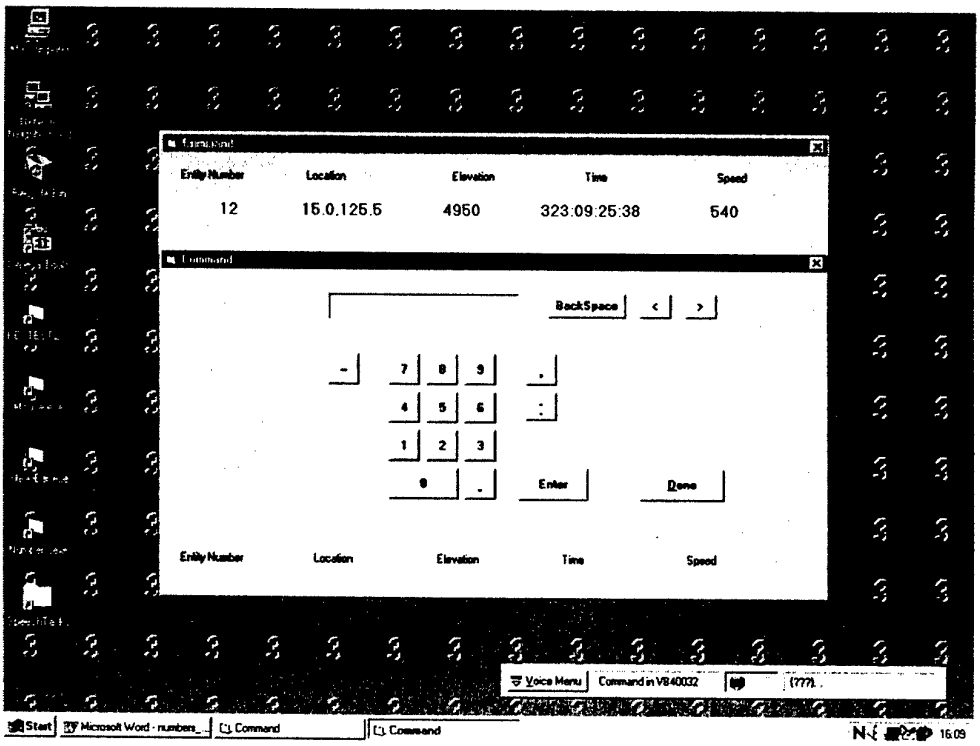


Figure 9: Numerical experimental interface.

The interface for all conditions except dictate mode ASR is shown in figure 9. The interface for the dictate mode condition differed only by the addition of the choice list

(like that in Figure 1) which was located to the left of the main window. The data to be entered was displayed in the higher windows and each field of data was copied into the long field at the top of the lower window. When the participant indicated that all the characters for a particular field had been entered correctly into the long window the characters were transferred for display in the corresponding field at the bottom of the window. Focus was shown by yellow shading of a field on the lowest line indicating to which field data would be transferred for display. Focus automatically moved on the assumption that data would be entered from left to right. Facilities for navigation within and between fields for the purpose of error correction varied between the conditions.

The stimuli were quintuplets of numbers intended to mimic aircraft track data so that some included decimal points, separators, and negative signs (in the latitude/longitude field). The shortest field held a two-digit number and the longest a twelve-symbol entry of nine digits and three special characters. There were equal representations of the digits 1-9 with a greater proportion of the digit 0 because of its increased frequency in real track data due to rounding.

5.2 ASR training

The vocabulary for the command mode ASR condition consisted of the digits 0-9, the commands "enter", "next", "back", "done", "backspace" "left", "right", and the special characters "comma", "point", "negative", "semi-colon" and "colon". Each participant trained the recogniser on this vocabulary by repeating each item 3-5 times. If the participant preferred to use "minus" rather than "negative", for example, then preferred term was trained in both the command and dictate conditions.

The vocabulary used in the dictate mode ASR condition was the same as that for the command mode condition with the addition of the two digit numbers, however this was a small subset of the total vocabulary available. Because the recogniser uses phonemic templates, it was not necessary to train all of the subset used. For instance, the template for "seventy-four" could be completed with the phonemes used in "seventy" and "four". On average participants trained twenty of the ninety two-digit numbers.

In dictate mode, the recogniser routinely took any two digit number to mean a numeral rather than a word. When each single digit numbers was first used it was recognised as a word rather than a numeral but when this was corrected using the choice list the recogniser nominated the numeral before the word on subsequent occasions.

5.3 Data entry and correction

The task consisted of entering a quintuplet of data using one of the four input modes. A new data quintuplet was displayed by pressing the Enter key. In the keypad condition, entry of the data in Figure 9, for instance, would be by the key sequence:
1 2 Enter
1 5 . 0 , 1 2 5 . 5 Enter
4 9 5 0 Enter
3 2 3 Shift+: 0 9 Shift+: 2 5 Shift+: 3 8 Enter
5 4 0 Enter
Ctrl+d

If necessary, navigation within the data entry field was by use of the left and right arrow keys and error correction was by use of the Backspace key and retyping. If necessary, navigation between fields was by use of the Enter or Tab key to move the focus forward in a cyclic fashion (or Shift+Tab to move it backwards) with the contents of the focus field at the bottom of the window being transferred back to the entry field for correction.

Button sequences in the mousepad condition were identical to the keying sequences in the keypad condition except that to enter ":" only a single click was necessary and the Done button substituted for the final Ctrl+d. Navigation within a field and between fields and error correction could be accomplished with the mousepad arrows, Enter, and BackSpace buttons but the mouse pointer could also be used directly to change focus between fields and to position the cursor within a field.

In the command mode condition, data was entered one character at a time so that, for instance, the quintuplet in Figure 9 would require 37 utterances. "Enter", "done", and "backspace" performed the same functions as the corresponding buttons on the mousepad, although back deletion of a single character could also be achieved with the standard recogniser command "scratch that". "Next" and "back" allowed navigation between fields while "left" and "right" allowed navigation within the entry field.

In the dictate mode condition, the quintuplet in Figure 9 would usually be input as the 28 utterance sequence:
"twelve" "enter"
"fifteen" "point" "zero" "comma" "twelve" "five" "point" "five" "enter"
"forty-nine" "fifty" "enter"
"thirty-two" "three" "colon" "zero" "nine" "colon" "twenty-five" "colon" "thirty-eight" "enter"
"fifty-four" "zero" "enter"
"done".
Alternatives such as "one, twenty-five" were possible instead of "twelve, five" for 125 and were demonstrated by the experimenter. Navigation and correction were achieved in the same way as in command mode except that "scratch that" would delete a

complete utterance which might be a two digit number in this condition and that the "left" and "right" commands were not available for navigation within a field.

5.4 Comparing performance by the four different modes of data entry

Figure 10 shows the average time to enter correct and corrected items by the four different modes employed in this experiment. In each mode, there is a noticeable speed increase from the first block of ten items to the second block. These learning effects are explained by the probability that all modes were somewhat novel to the participants. That is, by the time that they did this experiment they had a little experience with ASR and probably only small amounts of previous experience with the numerical keypad and with screen-based "desk accessory" calculators. The same learning effects can be seen in Appendix B in the data from the individual fields in each of the data entry conditions.

The ratio of the average time for a correct entry to the average time for a corrected entry is quite low in all conditions, even where ASR was employed. The contrast between these ratios and the much greater ratios in Figure 4 is explained by the lower proportion of corrections in each corrected entry in this experiment. That is, correct completion of an item in this experiment required over thirty correctly recognised utterances compared with only about three for an item in the Phrases Experiment. Many of the corrected items in both experiments required only one utterance to be repeated but the time for that stands out much more clearly in Figure 4 when averaged in with the times for two correctly recognised utterances than in Figure 10 when averaged in with the times for more than thirty correctly recognised utterances. The ratio is much larger, of course, in the statistics for individual fields (Appendix B) because of the smaller number of entries required to complete each field.

Another consequence of the large number of opportunities for error when entering a single item in this experiment is large error rates in all conditions. The proportions of items entered correctly first time in the second block of the ASR Command Mode, ASR Dictate Mode, Keypad, and Mousepad conditions were 32%, 22%, 65%, and 69% respectively. (Error rates are, of course, much lower in the data for individual fields.) As discussed above in section 4.3, these are not of great interest because of the small amount of training employed.

A particularly interesting comparison is that between ASR Command Mode and ASR Dictate Mode because Dictate Mode entry of each field should require fewer utterances than Command Mode (see Section 5.3). While the lower error rate in Command Mode is not surprising because of the smaller vocabulary considered by the recogniser as a possible match for each utterance, its overall speed advantage on correctly entered items is unexpected. In fact, Dictate Mode was only quicker for the third field of each item which always decomposed neatly into two two-digit numbers, thus requiring only two utterances as compared to the four required to enter the same number in

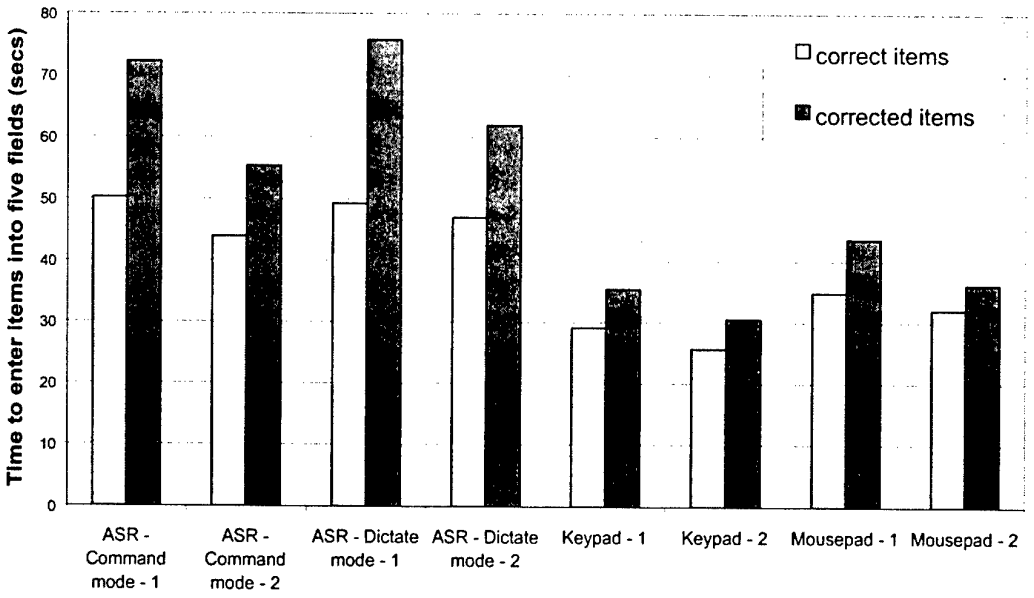


Figure 10: Average times to enter a quintuplet of numerical data, correctly and with corrections, in each of two blocks in each of the four conditions of the Numbers Experiment.

Command Mode. In the other fields, it is probable that when using Dictate Mode participants spent more "thinking time" parsing each entry into two-digit numbers than they saved by making fewer utterances and, in any case, those two-digit utterances would also have taken longer to enunciate than the single digit ones used in Command Mode. In so far as it was tested in this experiment, Dictate Mode does not appear to offer any advantage over Command mode for entry of numerical data, either for speed or accuracy.

The most interesting feature of Figure 10 is that correct numerical data entry is noticeably quicker by keypad or mousepad than by ASR. In fact, keypad entry was quicker than ASR Command Mode entry for almost every participant on every one of the five fields, rendering unnecessary an analysis of keying speed versus ASR entry speed similar to that in Figure 6 for the Phrases Experiment. This stands in contrast to the results from the other experiments in which ASR was seen to be comparable in speed to the other modes when errors were avoided. The explanation lies in the special nature of the data to be entered in this experiment. More precisely, it lies in the provision of a keypad and a mousepad designed specifically for the type of data to be entered. For instance, it is quicker for almost anyone to hit two keys '5' and '7' on the numerical keypad (or two screen buttons on the mousepad) than to say "five, seven" or "fifty-seven". This result generalises to the final rule of thumb that any keyboard or screen/mouse input interface purpose-built to match the data to be entered would be quicker for data entry than the equivalent ASR interface.

6. Discussion

6.1 Errors and error correction: recommendations for future research

In section 1, it was explained that the purpose of the current experiments was to compare rates of error-free, form-filling data entry using speaker-dependent, isolated-word ASR with "traditional" means of entering the same data. However as a result of the experiments, a few general observations are worth stating about the occurrence of errors and the correction of errors with current ASR technology.

- New users can expect very high error rates unless the required vocabulary is known in advance and trained in command mode.
- Error rates will fall very quickly, at least in the early stages.
- New users will initially find the error correction procedures very time-consuming but they can expect a mutual adaptation between themselves and the recogniser so that they are more often involved in only the simpler forms of correction (choosing the correct alternative from a list of possibilities rather than repeating a word and having to train the recogniser on that word).

The point of these observations is that errors and error correction should not be regarded as an insurmountable obstacle to the use of ASR for data entry by a regular operator who has trained the recogniser to his or her own voice, particularly when the task vocabulary is small enough to be trained in advance, the audio environment is stable and data entry is conducted in command mode. A good example of such an application can be found in Hashemi-Sahktsari, Broughton, and Martin (1999) where a vocabulary of 193 words and phrases was trained and used for communication with a computerised decision aid.

Error correction presents a much more significant problem when the users cannot train the complete vocabulary in advance. This may be either because the vocabulary is too large, not known in advance or because there is a large and variable user population or environment. In the latter case, technical advances are required in speaker independent systems which are currently only capable of recognising a small vocabulary. In the former case, further research is required specifically to measure the decline in error rates and the improvement in error correction procedures as users and recogniser adapt to each other. A different design from the current experiments would be required, one in which participants' interactions with the recogniser were monitored regularly over an extended period. (As a starting point, error correction could be examined in the raw data from the experiments reported here but since they do not explicitly differentiate between the use of different methods of error correction in different circumstances only limited insights could be gained.)

6.2 When ASR is worth considering in interface design on grounds of speed: conclusions from the current experiments

The outcome of each experiment was a rule of thumb to suggest when current, speaker-dependent, ASR technology may be an alternative worth considering on the grounds that it may be as quick as or quicker than the "traditional" alternative for data entry. Each heuristic carries the rider that the operator must have invested sufficient time in training the recogniser for data entry to be virtually error-free.

- Isolated word, dictate mode ASR is quicker than typing free text if the operator is a slow typist (<45 wpm).
- Command mode ASR is quicker than mouse selection from an on-screen list if the list is longer than 15 items.
- For numerical data, keypad or mousepad entry is quicker than ASR for almost all users

The second heuristic points to the most obviously fruitful area for ASR application in data entry which is nicely illustrated in Hashemi-Sahktsari, Broughton, and Martin (1999). However, it must be noted that the comparison in these experiments was only made with direct mouse selection from a list fully displayed on-screen. Other interface paradigms have been developed for selection from longer lists, the most common of which is a selection list which automatically scrolls as each letter of the sought word is typed into the search field. In a particular application involving selection from one or more lists, the interface designer has to consider whether the users will remember all possible entries and, if not, an on-screen or scrolling list may be required for browsing. Such considerations complicate the simple comparisons based only on speed of data entry where no "thinking time" is involved.

Numerical data is a special case where only a small number of keys (or mouse buttons) are used for each datum (and only a small set in total for all data) and where entry of that datum by ASR requires a similar number of separate utterances. Figure 10 illustrates that the average time to complete an utterance (most of which were monosyllabic) is more than 50% longer than the average time to hit a key (or mouse button). This suggests that it will usually be possible to design a special purpose data entry interface for each particular application which allows faster data entry than by use of ASR.

6.3 Adjusting for recogniser latency as computers get faster: will the conclusions remain relevant?

The heuristics summarised in section 6.2 are based on measurements made as the experimental participants used the hardware and software described in section 2.4. While it is probable that the participants are broadly representative of current and future users of data entry systems, it is also probable that the hardware and software will continue to be superseded by faster and faster versions. In this section we attempt

to estimate how the heuristics should be modified to allow for such improvements in the technology.

There is a slight initial increase in the speed of participants' correct entry of data by all modes. This can be seen clearly in of Figure 10 where the second "correct" bar for each condition is shorter than the first "correct" bar for the same condition. This is a practice effect. In the case of data entry by ASR, the participants learn to anticipate the response time (latency) of the recogniser (cf Wickens, 1992, p315). That is, they learn when to expect to see the last utterance printed on the screen and to proceed immediately to the next utterance. (The effect will be disrupted if the last utterance is not correctly recognised.) Figure 5 shows that the effect asymptotes out very quickly. Therefore, using the second "correct" bar from Figure 10 (or the corresponding data from Appendix B), we see that with practice it takes an average of 45.4s to speak the necessary utterances to the recogniser to enter a quintuplet of data in the ASR command mode condition of that experiment. From section 5.3 we know that each quintuplet requires about 37 utterances, almost all of which are monosyllabic. As a first approximation, we conclude (by division) that it takes about 1200ms to utter one syllable and have it correctly recognised.

Of this 1200ms, we know that the recogniser takes at most 400ms to recognise the utterance (Section 2.4). Put another way, if the technology advanced to the point where the utterance was recognised instantaneously, the cycle would still take about 800ms, a speed up of about 33%. A two syllable utterance adds 300-500ms to the cycle so that the maximum improvement of 400ms would mean a speed up of about 25%. A three-syllable utterance could be processed about 20% more quickly and so on.

Including the final "done", the words which had to be recognised in the Phrases Experiment averaged about two syllables each. So, by the reasoning in the previous paragraph, we would anticipate no more than a 25% speed up in processing if recogniser latency was eliminated. Referring to Figure 6, that would reduce the time to process a phrase from 6s to 4.5s, moving the crossover point of the two graphs up to a typing speed of 55wpm.

Including the final "done", the utterances which had to be recognised in the Lists Experiment also averaged about two syllables each, so that a maximum 25% speed up might be expected. Referring to Figure 8, that would reduce the time to process a data triplet from 6s to 4.5s, moving the crossover point of the two graphs down to a list length of zero. That is, if recogniser latency can be completely eliminated, faster data entry should be possible by error-free ASR than by mouse selection, regardless of list length.

The utterances which had to be recognised in the Numbers Experiment were mostly monosyllabic, so that a maximum 33% speed up might be expected. Referring to the second "correct' bar of Figure 10, that would reduce the average time to process a data

quintuplet from 44s to 29s. However, that is still longer than the average 26s achieved by participants on the second block of stimuli in the keypad condition.

It seems, then, that the broad conclusions of the previous section can be expected to hold even as hardware and isolated word ASR software become faster to the point where recogniser latency disappears. Special purpose interfaces (such as the numerical keypad) in which the number of keystrokes to enter a datum is no more than the number of utterances required to enter the same datum by ASR will remain faster than ASR, although the gap will close. The typing speed required to outpace ASR on free text will rise somewhat but will still be within the range commonly achieved by trained typists. The data entry application in which ASR is likely to be most successful will continue to be command mode selection from a finite list of alternatives.

7. Acknowledgements

We would like to thank Ahmad Hashemi-Sakhtsari for his thoughts, which stimulated us to design these experiments. We would also like to thank the staff of Human Systems Integration Group for their time and comments during the pilot study, especially Conn Copas and John Hansen. We wish to thank Mike Coleman for his effort in producing the software that was used. His efforts were greatly appreciated. Last but not least we would like to thank the volunteer participants.

References

Damper, R.I. & Wood, S.D. (1995). Speech versus keying in command and control applications. International Journal of Human-Computer Studies, 42, 289-305.

Hashemi-Sahktsari, A., Broughton, M. and Martin, C. (1999). Application of automatic speech recognition to a Sonar Operator Decision aid. Defence Science and Technology Organisation Client Report, DSTO-CR-0103.

Karl, L.R., Pettey, M. & Shneiderman, B. (1993). Speech versus mouse commands for word processing: an empirical evaluation. International Journal of Man-Machine Studies, 39, 667-687.

Leggett, J. & Williams, G. (1984). An empirical investigation of voice as an input modality for computer programming. International Journal of Man-Machine Studies, 21, 493-520.

McSorley, W.J. (1981, March). Using voice recognition equipment to run the Warfare Environmental Simulator (WES). Unpublished Masters thesis, Naval Postgraduate School, Monterey, CA.

Morrison, D.L., Green, T.R.G., Shaw, A.C. & Payne, S.J. (1984). Speech controlled text-editing: effects of input modality and of command structure. International Journal of Man-Machine Studies, 21, 49-63.

Simpson, C.A., McCauley, M.E., Roland, E.F., Ruth, J.C. & Williges, B.H. (1985). System design for speech recognition and generation. Human Factors, 27(2), 115-141.

Suhm, B., Myers, B. & Waibel, A. (1996). Interactive recovery from speech recognition errors in speech user interfaces. Proceedings ICSLP 96. Fourth International Conference on Spoken Language Processing, V2, 865-868.

Welch, J.R. (1977, September). Automated data entry analysis (RADC TR-77-306). Griffiss AFB, New York: Rome Air Development Center.

Wickens, C. D. (1992). Engineering psychology and human performance. New York: Harper-Collins.

Appendix A: Phrase vocabulary

AIRCRAFT FROM BASE
FIGHTER FROM SQN
SQN FROM BASE
PILOT FROM BASE
FROM TAOC TO FA18
FROM TAOC TO SADOC
FROM TAOC TO SQN
FROM TAOC TO PILOT
FRIENDLY
HOSTILE
NEUTRAL
UNKNOWN
FRIENDLY AIRCRAFT
HOSTILE AIRCRAFT
NEAREST AIRCRAFT
NEUTRAL AIRCRAFT
FRIENDLY FIGHTER
HOSTILE FIGHTER
NEAREST FIGHTER
NEUTRAL FIGHTER
UNKNOWN FIGHTER
ACTIVE FIGHTER
FIGHTER DOWN
INTERCEPTION ROUTE
INTERCEPTION COURSE
CONVERGENCE COURSE
ENCOUNTER PATH
CONVERGENCE TRACK
INTERCEPTION TRAJECTORY
INTERCEPT PATH
NEW ROUTE REQUIRED
NEW COURSE REQUIRED
FIRST PATH
NEW TRAJECTORY
INITIAL FLIGHT PATH
NEW GLIDE PATH
INITIAL TRACK
INITIAL COURSE
FIRST COURSE
NEW TRACK
FIRST TRAJECTORY
CHANGE ROUTE AGAIN
MODIFY ROUTE
FURTHER ROUTE CHANGE
ANOTHER ROUTE CHANGE
PATH ALTERATION
SWITCH TRAJECTORY
AMEND COURSE
REVISE TRACK
MODIFY PATH
SWITCH COURSE
TRACK AMENDMENT
ALTERED COURSE
CHANGE PATH AGAIN
TRAJECTORY CHANGE
NEXT PATH
REVERSE COURSE
RETURN ROUTE TO BASE
RETURN COURSE TO SQN
PATH BACK TO BASE
REVERSE PATH TO BASE
REVERSE COURSE
RETURN HOME
RETREAT TRACK
RETREAT PATH
COURSE WITHDRAWN
RETURN TRACK
TRACK TO BASE
REVERSE ROUTE
NEAREST AIRCRAFT
NEAREST BASE
NEAREST SQN
NEAREST FIGHTER
CLOSEST AIRCRAFT
CLOSEST BASE
CLOSEST SQN
CLOSEST FIGHTER
APPROACHING AIRCRAFT
APPROACHING
BASE
APPROACH SQN
APPROACHING FIGHTER

IMMEDIATE AIRCRAFT
IMMEDIATE BASE
IMMEDIATE SQN
IMMEDIATE FIGHTER
NEARBY AIRCRAFT
NEARBY BASE
NEARBY SQN
NEARBY FIGHTER
TARGET LOCATION
TARGET PATH
TARGET CHANGE
INTERCEPT TARGET
JOIN FORMATION
LEAVE FORMATION
HOSTILE TARGET
UNKNOWN TARGET

Appendix B: Statistics on the raw experimental data

Statistics are given on the number of observations (data points), for the means, and for the spreads of times for correct and corrected entries within each experimental condition. Times for incorrect or incomplete entries are omitted. The mean and spread statistics weight each observation equally so that, for instance, a participant who made six correct entries in a particular condition contributes more to the mean and standard deviation of correct entry times for that condition than does a participant who made only five correct entries. The purpose of this approach is to identify outlier observations for correct and corrected entries in each condition.

B.1 Data from Lists Experiment

LISTS EXPERIMENT	Data entry condition				
	Mouse selection			ASR	
	25 item list	10 item list	5 item list	25 item list	5 item list
Number of observations	480	480	480	480	480
Number of correct entries	447	459	464	331	341
Mean time for correct entries (secs)	7.41	5.79	5.29	6.65	6.38
Standard deviation of times for correct entries	1.79	1.26	0.98	3.53	4.62
Number of correct entries taking longer than 3 sd from the mean	5	4	4	3	2
Number of correct entries taking longer than 4 sd from the mean	1	0	1	3	2
Number of correct entries taking longer than 5 sd from the mean	0	0	1	3	2
Number of corrected entries	32	19	16	114	109
Mean time for corrected entries (secs)	7.40	5.55	6.24	21.69	18.67
Standard deviation of times for corrected entries	1.66	0.74	1.55	29.10	18.15
Number of corrected entries taking longer than 3 sd from the mean	1	0	1	1	4
Number of corrected entries taking longer than 4 sd from the mean	0	0	0	1	2
Number of corrected entries taking longer than 5 sd from the mean	0	0	0	1	1

B.2 Data from Phrases Experiment

PHRASES EXPERIMENT	Data entry condition				
	Keyboard	ASR			
		Block 1	Block 2	Block 3	Block 4
Number of observations	456	432	432	432	432
Number of correct entries	330	133	250	279	329
Mean time for correct entries (secs)	7.12	7.76	7.31	7.53	6.15
Standard deviation of times for correct entries	2.96	4.54	3.39	5.15	1.82
Number of correct entries taking longer than 3 sd from the mean	5	2	2	3	3
Number of correct entries taking longer than 4 sd from the mean	3	2	2	3	2
Number of correct entries taking longer than 5 sd from the mean	0	2	2	3	2
Number of corrected entries	107	283	174	146	93
Mean time for corrected entries (secs)	9.23	41.36	28.58	23.49	22.07
Standard deviation of times for corrected entries	3.80	40.15	27.55	20.13	18.40
Number of corrected entries taking longer than 3 sd from the mean	1	4	3	2	2
Number of corrected entries taking longer than 4 sd from the mean	0	3	1	2	0
Number of corrected entries taking longer than 5 sd from the mean	0	2	1	0	0

B.3 Data from Numbers Experiment

B.3.1 Data from field 1 in Numbers Experiment

NUMBERS EXP - FIELD 1	Data entry condition							
	ASR Command Mode		ASR Dictate Mode		Keypad		Mousepad	
	Block 1	Block 2	Block 1	Block 2	Block 1	Block 2	Block 1	Block 2
Number of observations	240	240	240	240	240	240	240	240
Number of correct entries	194	212	185	187	228	229	235	225
Mean time for correct entries (secs)	5.08	4.90	5.60	6.07	4.09	3.37	3.91	3.45
Standard deviation of times for correct entries	2.57	2.52	2.34	5.53	1.94	1.28	2.25	1.69
Number of correct entries taking longer than 3 sd from the mean	2	3	5	1	2	2	1	2
Number of correct entries taking longer than 4 sd from the mean	2	3	2	1	1	1	1	2
Number of correct entries taking longer than 5 sd from the mean	1	2	2	1	1	1	1	1
Number of corrected entries	38	25	54	44	7	7	5	5
Mean time for corrected entries (secs)	14.72	23.19	26.94	13.43	6.46	8.54	17.47	7.28
Standard deviation of times for corrected entries	13.57	46.42	74.66	4.71	1.81	3.88	13.64	1.19
Number of corrected entries taking longer than 3 sd from the mean	1	1	1	1	0	0	0	0
Number of corrected entries taking longer than 4 sd from the mean	1	1	1	0	0	0	0	0
Number of corrected entries taking longer than 5 sd from the mean	0	0	1	0	0	0	0	0

B.3.2 Data from field 2 in Numbers Experiment

NUMBERS EXP - FIELD 2	Data entry condition							
	ASR Command Mode		ASR Dictate Mode		Keypad		Mousepad	
	Block 1	Block 2	Block 1	Block 2	Block 1	Block 2	Block 1	Block 2
Number of observations	240	240	240	240	240	240	240	240
Number of correct entries	161	169	121	177	201	203	210	210
Mean time for correct entries (secs)	15.88	13.61	16.17	14.84	9.36	8.18	11.13	10.51
Standard deviation of times for correct entries	6.69	5.79	3.04	2.54	3.28	2.92	2.76	2.53
Number of correct entries taking longer than 3 sd from the mean	3	1	3	2	2	2	1	1
Number of correct entries taking longer than 4 sd from the mean	3	1	2	1	2	1	1	0
Number of correct entries taking longer than 5 sd from the mean	1	1	1	1	2	1	0	0
Number of corrected entries	64	66	116	50	37	33	19	10
Mean time for corrected entries (secs)	32.64	32.46	38.49	24.33	16.92	12.92	16.12	14.03
Standard deviation of times for corrected entries	25.86	68.63	33.64	6.20	9.92	7.62	3.64	2.93
Number of corrected entries taking longer than 3 sd from the mean	2	1	1	0	1	1	0	0
Number of corrected entries taking longer than 4 sd from the mean	1	1	1	0	0	0	0	0
Number of corrected entries taking longer than 5 sd from the mean	0	1	1	0	0	0	0	0

B.3.3 Data from field 3 in Numbers Experiment

NUMBERS EXP – FIELD 3	Data entry condition							
	ASR Command Mode		ASR Dictate Mode		Keypad		Mousepad	
	Block 1	Block 2	Block 1	Block 2	Block 1	Block 2	Block 1	Block 2
Number of observations	240	240	240	240	240	240	240	240
Number of correct entries	194	199	205	208	234	236	231	227
Mean time for correct entries (secs)	6.97	6.25	5.98	5.23	2.90	2.79	4.37	3.68
Standard deviation of times for correct entries	2.52	1.42	2.66	1.02	1.26	1.36	1.47	0.99
Number of correct entries taking longer than 3 sd from the mean	3	5	2	5	4	2	8	5
Number of correct entries taking longer than 4 sd from the mean	2	2	2	3	2	1	1	1
Number of correct entries taking longer than 5 sd from the mean	2	1	1	1	2	1	0	1
Number of corrected entries	30	36	34	20	5	4	8	2
Mean time for corrected entries (secs)	19.82	12.57	18.82	14.57	10.93	4.66	19.54	4.87
Standard deviation of times for corrected entries	15.22	6.88	14.34	6.11	7.34	1.68	18.69	0.35
Number of corrected entries taking longer than 3 sd from the mean	0	1	1	0	0	0	0	0
Number of corrected entries taking longer than 4 sd from the mean	0	1	1	0	0	0	0	0
Number of corrected entries taking longer than 5 sd from the mean	0	1	0	0	0	0	0	0

B.3.4 Data from field 4 in Numbers Experiment

NUMBERS EXP - FIELD 4	Data entry condition							
	ASR Command Mode		ASR Dictate Mode		Keypad		Mousepad	
	Block 1	Block 2	Block 1	Block 2	Block 1	Block 2	Block 1	Block 2
Number of observations	240	240	240	240	240	240	240	240
Number of correct entries	144	155	95	126	209	207	220	201
Mean time for correct entries (secs)	17.96	15.90	17.40	15.88	10.83	9.51	12.12	11.63
Standard deviation of times for correct entries	7.13	5.31	2.85	3.10	3.31	2.75	2.80	2.80
Number of correct entries taking longer than 3 sd from the mean	4	2	0	1	1	2	4	3
Number of correct entries taking longer than 4 sd from the mean	1	2	0	1	1	0	2	2
Number of correct entries taking longer than 5 sd from the mean	1	1	0	1	0	0	0	0
Number of corrected entries	74	81	142	97	29	33	16	24
Mean time for corrected entries (secs)	38.56	24.24	32.69	28.60	15.28	11.00	18.50	16.34
Standard deviation of times for corrected entries	38.75	13.83	22.54	11.54	5.33	3.20	5.07	5.02
Number of corrected entries taking longer than 3 sd from the mean	1	1	6	1	1	0	0	1
Number of corrected entries taking longer than 4 sd from the mean	1	1	2	1	0	0	0	1
Number of corrected entries taking longer than 5 sd from the mean	1	1	1	1	0	0	0	0

B.3.5 Data from field 5 in Numbers Experiment

NUMBERS EXP - FIELD 5	Data entry condition							
	ASR Command Mode		ASR Dictate Mode		Keypad		Mousepad	
	Block 1	Block 2	Block 1	Block 2	Block 1	Block 2	Block 1	Block 2
Number of observations	240	240	240	240	240	240	240	240
Number of correct entries	179	211	166	175	235	229	231	221
Mean time for correct entries (secs)	6.27	5.25	5.63	5.28	2.60	2.39	3.60	3.42
Standard deviation of times for correct entries	4.60	5.06	0.97	1.42	1.25	0.92	1.02	1.20
Number of correct entries taking longer than 3 sd from the mean	4	1	1	1	2	4	4	2
Number of correct entries taking longer than 4 sd from the mean	4	1	0	1	2	0	0	1
Number of correct entries taking longer than 5 sd from the mean	3	1	0	1	2	0	0	1
Number of corrected entries	30	23	70	48	5	8	2	3
Mean time for corrected entries (secs)	17.30	10.25	11.81	14.15	17.17	4.11	6.10	8.57
Standard deviation of times for corrected entries	12.92	2.71	6.40	11.13	21.00	0.88	0.71	1.83
Number of corrected entries taking longer than 3 sd from the mean	0	0	2	2	0	0	0	0
Number of corrected entries taking longer than 4 sd from the mean	0	0	1	1	0	0	0	0
Number of corrected entries taking longer than 5 sd from the mean	0	0	0	0	0	0	0	0

B.3.6 Data from all fields in Numbers Experiment

NUMBERS EXP - ALL FIELDS	Data entry condition							
	ASR Command Mode		ASR Dictate Mode		Keypad		Mousepad	
	Block 1	Block 2	Block 1	Block 2	Block 1	Block 2	Block 1	Block 2
Number of observations	240	240	240	240	240	240	240	240
Number of correct entries	72	82	34	53	164	158	179	172
Mean time for correct entries (secs)	50.11	44.80	48.72	47.47	29.98	26.27	35.73	32.83
Standard deviation of times for correct entries	10.15	9.49	5.31	10.94	8.23	6.36	6.99	6.48
Number of correct entries taking longer than 3 sd from the mean	2	2	1	1	1	0	2	1
Number of correct entries taking longer than 4 sd from the mean	0	1	0	1	1	0	1	1
Number of correct entries taking longer than 5 sd from the mean	0	1	0	1	1	0	0	0
Number of corrected entries	134	149	200	164	67	72	42	39
Mean time for corrected entries (secs)	75.52	64.01	84.52	63.10	35.92	29.06	42.89	36.26
Standard deviation of times for corrected entries	41.25	52.51	59.63	14.22	14.44	8.52	13.10	5.67
Number of corrected entries taking longer than 3 sd from the mean	4	2	3	4	2	0	1	1
Number of corrected entries taking longer than 4 sd from the mean	1	2	3	1	0	0	1	0
Number of corrected entries taking longer than 5 sd from the mean	1	1	2	0	0	0	0	0

Appendix C: Statistics on the data used for the report

Statistics are given on the number of observations (data points) and for the means of times for correct and corrected entries within each experimental condition. Times for incorrect or incomplete entries and outliers more than four standard deviations from the relevant mean are omitted. (See Appendix A.) The means weight each participant equally so that, for instance, a participant who made six correct entries in a particular condition contributes no more to the mean of correct entry times tabulated for that condition than does a participant who made only five correct entries.

LISTS EXPERIMENT	Data entry condition				
	Mouse selection			ASR	
	25 item list	10 item list	5 item list	25 item list	5 item list
Number of observations	480	480	480	480	480
Number of correct entries	446	459	463	328	339
Mean time for correct entries (secs)	7.35	5.75	5.26	6.42	6.09
Number of corrected entries	32	19	16	113	107
Mean time for corrected entries (secs)	7.85	5.50	6.52	17.39	17.87

PHRASES EXPERIMENT	Data entry condition				
	Keyboard	ASR			
		Block 1	Block 2	Block 3	Block 4
Number of observations	456	432	432	432	432
Number of correct entries	327	131	248	276	327
Mean time for correct entries (secs)	6.81	7.28	7.09	7.10	6.10
Number of corrected entries	107	280	173	144	93
Mean time for corrected entries (secs)	10.04	38.51	27.96	22.62	21.40

NUMBERS EXP	Data entry condition							
	ASR Command Mode		ASR Dictate Mode		Keypad		Mousepad	
	Block 1	Block 2	Block 1	Block 2	Block 1	Block 2	Block 1	Block 2
Number of observations	240	240	240	240	240	240	240	240
FIELD 1								
Number of correct entries	192	209	183	186	227	228	234	223
Mean time for correct entries (secs)	4.84	4.70	5.50	5.66	4.00	3.36	3.79	3.33
Number of corrected entries	37	24	53	44	7	7	5	5
Mean time for corrected entries (secs)	12.83	14.74	16.72	13.47	6.11	8.54	17.47	7.28
FIELD 2								
Number of correct entries	158	168	119	176	199	202	209	210
Mean time for correct entries (secs)	15.16	13.27	16.05	14.70	9.20	8.03	10.92	10.46
Number of corrected entries	63	65	115	50	37	33	19	10
Mean time for corrected entries (secs)	33.06	25.00	36.60	23.01	17.45	13.41	15.46	14.55
FIELD 3								
Number of correct entries	192	197	203	205	232	235	230	226
Mean time for correct entries (secs)	6.75	6.23	5.76	5.15	2.84	2.72	4.35	3.65
Number of corrected entries	30	35	33	20	5	4	8	2
Mean time for corrected entries (secs)	21.47	11.40	16.80	14.19	10.93	4.66	20.05	4.87
FIELD 4								
Number of correct entries	143	153	95	125	208	207	218	199
Mean time for correct entries (secs)	17.82	15.57	17.46	15.61	10.73	9.41	12.01	11.45
Number of corrected entries	73	80	140	96	29	33	16	23
Mean time for corrected entries (secs)	35.49	22.33	31.26	28.21	14.32	11.09	19.31	15.50
FIELD 5								
Number of correct entries	175	210	166	174	233	229	231	220
Mean time for correct entries (secs)	5.66	4.95	5.64	5.23	2.54	2.39	3.59	3.38
Number of corrected entries	30	23	69	47	5	8	2	3
Mean time for corrected entries (secs)	16.74	10.01	11.63	12.27	12.97	4.18	6.10	8.57
ALL FIELDS								
Number of correct entries	72	81	34	52	163	158	178	171
Mean time for correct entries (secs)	51.19	45.38	49.99	47.01	29.29	25.78	34.93	32.30
Number of corrected entries	133	147	197	163	67	72	41	39
Mean time for corrected entries (secs)	74.79	57.30	78.72	62.65	37.06	30.71	42.13	36.64

Appendix D: Enrolment vocabulary

Command Mode
Dictate Mode
Go to sleep
Help
Lessons
Next slide
Oops
Previous slide
Speed
Wake up
What can I say
International
Recognition
Technology
To
The

DISTRIBUTION LIST

Experimental Comparisons of Data Entry by Automatic Speech Recognition, Keyboard, and Mouse

Helen M. Mitchard and Jim Winkles

AUSTRALIA

DEFENCE ORGANISATION

Task Sponsor
DOIS(D)

S&T Program
Chief Defence Scientist
FAS Science Policy
AS Science Corporate Management
Director General Science Policy Development
(shared copy)
Counsellor Defence Science, London (Doc Data Sheet)
Counsellor Defence Science, Washington (Doc Data Sheet)
Scientific Adviser to MRDC Thailand (Doc Data Sheet)
Scientific Adviser Joint
Navy Scientific Adviser (Doc Data Sheet and distribution list only)
Scientific Adviser - Army (Doc Data Sheet and distribution list only)
Air Force Scientific Adviser
Director Trials

Aeronautical and Maritime Research Laboratory
Director

Electronics and Surveillance Research Laboratory
Chief of Information Technology Division (Doc Data Sheet & distribution list only)
Research LeaderCommand and Control & Intelligence Systems Branch
Authors: Helen M. Mitchard (24 copies) and Jim R. Winkles

DSTO Library and Archives
Library Fishermans Bend (Doc Data Sheet)
Library Maribyrnong (Doc Data Sheet)
Library Edinburgh
Australian Archives
Library, MOD,
Library, MOD, Pyrmont (Doc Data sheet only)
US Defense Technical Information Center, 2 copies
UK Defence Research Information Centre, 2 copies
Canada Defence Scientific Information Service, 1 copy
NZ Defence Information Centre, 1 copy
National Library of Australia, 1 copy

Capability Systems Staff
Director General Maritime Development (Doc Data Sheet only)
Director General Aerospace Development

Knowledge Staff
Director General Command, Control, Communications and Computers (DGC4) (Doc Data Sheet only)

Navy
SO (SCIENCE), COMAUSNAVSURFGRP, Bld 95, Garden Island, Locked Bag 12, Pyrmont NSW 2009 (Doc Data Sheet only)

Army
Stuart Schnaars, ABCA Standardisation Officer, Tobruk Barracks, Puckapunyal, 3662 (4 copies)
SO (Science), Deployable Joint Force Headquarters (DJFHQ) (L), MILPO Gallipoli Barracks, Enoggera QLD 4052 (Doc Data Sheet only)

Air Force
DOIS(D)

Intelligence Program
DGSTA Defence Intelligence Organisation
Manager, Information Centre, Defence Intelligence Organisation

Corporate Support Program
Library Manager, DLS-Canberra

UNIVERSITIES AND COLLEGES
Australian Defence Force Academy
Library
Head of Aerospace and Mechanical Engineering
Serials Section (M list), Deakin University Library, Geelong, 3217
Hargrave Library, Monash University (Doc Data Sheet only)
Librarian, Flinders University

OTHER ORGANISATIONS
NASA (Canberra)
AusInfo
State Library of South Australia

OUTSIDE AUSTRALIA

ABSTRACTING AND INFORMATION ORGANISATIONS
Library, Chemical Abstracts Reference Service
Engineering Societies Library, US
Materials Information, Cambridge Scientific Abstracts, US
Documents Librarian, The Center for Research Libraries, US

INFORMATION EXCHANGE AGREEMENT PARTNERS

Acquisitions Unit, Science Reference and Information Service, UK
Library - Exchange Desk, National Institute of Standards and Technology, US
National Aerospace Laboratory, Japan
National Aerospace Laboratory, Netherlands

SPARES (5 copies)

Total number of copies: **69 copies**

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
DOCUMENT CONTROL DATA

1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)

2. TITLE

Experimental Comparisons of Data Entry by Automatic Speech Recognition, Keyboard, and Mouse

3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)

Document	(U)
Title	(U)
Abstract	(U)

4. AUTHOR(S)

Helen Mitchard and Jim Winkles

5. CORPORATE AUTHOR

Electronics and Surveillance Research Laboratory
PO Box 1500
Edinburgh SA 5111 Australia

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DSTO-RR-0220	AR-012-066	Research Report	November 2001

8. FILE NUMBER	9. TASK NUMBER	10. TASK SPONSOR	11. NO. OF PAGES	12. NO. OF REFERENCES
9505-19-229	JNT 96/231	DOIS(D)	61	11

13. URL ON THE WORLDWIDE WEB

http://www.dsto.defence.gov.au/corporate/reports/DSTO-RR-0220.pdf

14. RELEASE AUTHORITY

Chief, Information Technology Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE,PO BOX 1500 EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CASUAL ANNOUNCEMENT Yes

18. DEFTEST DESCRIPTORS

Speech recognition
Voice data processing
Data processing
Input output devices (computing)
Message processing

19. ABSTRACT

The objective was to determine the conditions under which Automatic Speech Recognition (ASR) is an efficient choice for data entry. In particular the focus was on data entry tasks that are part of constructing military messages. The ADF Formatted Messaging System utilises a structured formatting system to constrain the semantics of a message but also includes a field for unlimited and unstructured text. Hence the data entry tasks involved range from form-filling to free dictation of short phrases. In the experiments, ASR and manual input modes are compared for three data entry tasks: textual phrase entry, selection from a list, and numerical data entry. To effect fair comparisons, the tasks minimised the transaction cycle for each input mode and data type and the main comparisons use only times from correct data entry. The results indicate that for inputting short phrases ASR only competes if the typist's speed is below 45wpm. For selecting an item from a list, ASR offered an advantage only if the list length was greater than 15 items. For entering numerical data, ASR offered no advantage over keypad or mouse. The general conclusion for formatted data entry is that a keyboard/mouse interface designed to match the data to be entered will be more time efficient than any equivalent ASR interface.