Manuscript accepted for publication in *International Journal of Human-Computer Studies*, originally submitted May 31, 1994.

Speech versus Keying in Command and Control Applications[†]

R.I. Damper and S.D. Wood,

Image, Speech and Intelligent Systems (ISIS) Group,

Department of Electronics and Computer Science,

University of Southampton,

Southampton SO17 1BJ,

UK.

[†]Based on a paper presented at joint European Speech Communication Association/NATO Research Study Group in Speech Processing (RSG10) Workshop on "Speech Technology Applications", Lautrach, Germany, September 1993.

Abstract

Experimental comparisons of speech and competitor input media such as keying have, taken overall, produced equivocal results: this has usually been attributed to "task-specific variables". Thus, it seems that there are some good, and some less good, situations for utilisation of speech input. One application generally thought to be a success is small-vocabulary, isolated-word recognition for command and control. In a simulated command and control task, Poock (1980; 1982) purportedly showed a very significant superiority of speech over keying in terms of higher input speeds and lower error rates. This paper argues that the apparent superiority observed results from a methodological error – specifically that the verbose commands chosen suit the requirements of speech input but make little or no concession to the requirements of keying. We describe experiments modelled on those of Poock, but designed to overcome this putative flaw and to effect a *fair comparison* of the input media by using terse, abbreviated commands for the keying condition at least. Results of these new experiments reveal that speech input is 10.6% slower (although this difference is not statistically significant) and 360.4% more error-prone than keying, supporting our hypothesis that the methodology of the earlier work was flawed. However, simple extrapolation of our data for terse commands to the situation where keyed commands are entered in full suggests that other differences between our work and Poock's could play a part. Overall, we conclude that a fair comparison of input media requires an experimental design that explicitly attempts to minimise the so-called transaction cycle – the number of user actions necessary to elicit a system response – for each medium.

List of Figures

List of Tables

Introduction

Several years of concentrated research effort have resulted in significant advances in the technology of speech input to computers. Most usually, this effort has been justified by an almost implicit assumption that automatic speech recognition (ASR) offers the key to dramatic improvements in the effectiveness of the human-computer interface, rather than by reference to any relevant human factors literature. For instance, in the landmark volume documenting the state-of-the-art in ASR at the end of the 1970s, Lea (1980) writes:

> "... you will want to use speech whenever possible because it is the human's most natural communication modality."

More recently, Lee (1989) in his highly regarded text states:

> "Voice input to computers offers ... a natural, fast, hands free, eyes free, location free input medium."

While the superiority of speech as an input medium is most often merely assumed, some authorities (e.g. Ainsworth, 1988, pp. 3–4) refer to the influential work of Chapanis (1975) and his co-workers (Chapanis, Parrish, Ochsman & Weeks, 1977) who have shown a clear advantage to the use of speech in cooperative problem solving between humans in terms of solution speed. However, there are obvious and marked differences between human-human and human-computer interaction which mean that advantages in the former case do not necessarily transfer to the latter situation (Amalberti, Carbonell & Falzon, 1993; p. 563).

Less often do authors attempt a more direct justification for ASR based on demonstrated, concrete advantages of speech relative to alternative input media. One

possible reason for this is provided by Karl, Pettey & Shneiderman (1993) who write:

> "Despite advances in speech technology, human factors research since the late 1970s has provided only weak evidence that ASR devices are superior to conventional input devices."

So, to what extent is the evidence "weak"? According to Karl *et al*, "... early studies taken as a whole are not conclusive". Summarising a comprehensive review of the literature, Martin (1989) states: "As these examples show, the results of formal comparisons between keyboard and speech are often contradictory and ambiguous". Earlier still, Simpson, McCauley, Roland, Ruth & Williges (1985, p. 121) write that experimental comparisons of the relative merits of speech input and more conventional means of data entry to computers have often "produced conflicting results, depending upon the unit of input (alphanumerics or functions) and other task-specific variables". In other words, the particular application and specific requirements of the interface design play an important part. While this must undoubtedly be so, another possible reason for such conflict could be the difficulty of deciding what constitutes a *fair comparison* between different input media.

Our purpose in this paper is to explore experimentally the inconclusive and conflicting results obtained to date, in order to gain a clearer view of the relative advantages of speech and keying. We concentrate on the issue of fair comparison between competitor media. Underlying the work is the notion that the human factors of speech input is a difficult area in which to work. Thus, it is entirely conceivable that early studies may inadvertently – through subtle methodological shortcomings, for instance – have portrayed speech in an unduly optimistic (or indeed pessimistic) light. Accordingly, it is necessary to appraise critically the methodology of previous

workers, rather than accepting their results at face value.

Our approach has been to identify a particular study, namely that of Poock (1980; 1982), which stands out in the extent of the advantage it apparently shows for speech over keying. In a simulated naval command and control task using a distributed computer network (ARPANET), Poock reports that speech input is somewhat faster and enormously less error-prone than keying. Careful scrutiny of the experimental design, however, reveals what we believe to be a significant methodological error introducing a bias favouring speech. Accordingly, we have performed new experiments modelled on Poock's work, but using a slightly different design intended to correct this perceived error. The point at issue is the effect this modification has on the results obtained and, thereby, on the view one obtains of the superiority (or otherwise) of speech over keying.

We are very conscious that we are focusing on a rather dated study – from more than 10 years ago. Hence, it could be argued that its results are no longer relevant. We would counter this argument on several grounds, not least that Poock's work has been highly influential. (See the Discussion section below for further arguments.) In those cases where a justification for the usability of speech input based on the human factors literature is attempted, his study figures prominently. For instance, Schurick, Williges & Maynard (1985) write:

> "Some successful applications of speech recognition include ... command and control (Poock, 1980)"

... while Karl *et al* (1993) state:

> "However, the advantage of voice input over keyboard for command activation (Poock, 1982) ... is clear".

Overall then, informed opinion holds that it is the specific task studied which accounts for Poock's optimistic results, and that other applications of ASR to command and control are very likely to be successful. It is our contention that this interpretation may be incorrect. If we are right, the relevance to present-day applications is obvious.

The remainder of the paper is structured as follows. We next review the literature on the experimental comparison of speech and keying. Because of our central concern with command and control, we restrict consideration to isolated-word, small-vocabulary ASR. This review parallels (and slightly updates) that presented by Martin (1989) but has a somewhat different perspective. We then give a critical analysis of Poock's (1980; 1982) work, before describing our own experiments. As stated above, these are broadly similar to those of Poock but with important differences designed to test the hypothesis that a methodological flaw explains the apparent superiority of speech over keying in his study. The results of these new experiments are outlined. We then discuss these results and how they differ from those obtained by Poock, before finally drawing conclusions relating to the issue of speech input versus keying in small-vocabulary applications (such as command and control) and to the conduct of future work on the human factors of speech input.

Speech versus Keying: a Review

The results of early work comparing speech and keying were essentially pessimistic with regard to the utility of speech input in isolated-word, small-vocabulary, primary data-entry tasks. Advantages were, however, more apparent in situations of concurrent, secondary tasking and high work-load. Using speech recognition, keying and light pen for primary data entry (the so-called "simple scenario"), Welch

(1977) found that the keyboard provided the fastest and most accurate entry of random numeric strings, although the percentage degradation in performance when a secondary, hand-occupation task was added was smaller for speech (10%) than for keying (30%). For random alphanumeric data entry in the simple scenario, keyboard was also faster but speech had a lower error rate.

Nye (1982; p. 54) describes attempts by United Airlines to replace keyed entry of airline baggage destination information with spoken entry. The established procedure was for one baggage handler to read aloud a 3-digit flight identifier code which was then keyed by a second operator. This code determined which of 32 sorter exit ramps was opened to allow the baggage through. As well as requiring two operators, unacceptably high error rates in the range 10-40% were encountered. This is a classic 'hands-busy' situation in which speech should offer an advantage by providing an additional, non-manual input channel. Accordingly, direct entry of the code spoken by the first operator, removing the need for a second operator, is attractive. However, Nye reports:

> "Over a five-year period, United tried a number of speech recognition machines instead of keyboard entry but was not able to lower the error rate appreciably."

Subsequently, it was decided to speak the name of the destination city, rather than the arbitrary code, and simply let the computer perform the encoding. The effect of this was dramatic in that the speech condition then showed a clear superiority with respect to the keyed condition. In a six-week trial involving 9 speakers (1 female), a misrecognition rate of just 0.4% was observed with an average rejection rate of 7%. Most of this error was attributed to "newer operators becoming familiar with the system" (performance improved with time). Clearly, meaningful names were easier

for the users to remember and say than abbreviated codes, which had been chosen at the outset because they were easier to *type*. Martin (1989; p. 357) draws an important lesson from this result, stating:

> "... simple comparisons of response channel efficiency may not always be the best measure of the potential of speech input. This is because the form of the input may change with different modalities."

In the study of Morrison, Green, Shaw & Payne (1984), subjects entered literal text by keyboard but edited it either by speech commands or by key-press. A possible advantage for speech-plus-keyboard over keyboard alone is advanced, namely that the former "achieves a classic separation of function by modality". Two different types of editor were employed: one required more but simpler commands (MC), the other required fewer but more complex commands (FC). Morrison *et al* introduce the notion of *transaction cycle*, i.e. the number of actions required before the system responded. Of their 20 subjects, none had ASR experience; 10 were expert typists and the remainder were non-typists. Each individual used only one of the MC or FC editors. Results showed that, regarding task completion: "No particular editor or input medium was faster to use overall". Considering command entry alone, however: "speech input was uniformly slower, probably because of our limited hardware". Error rates were comparable across media. Typists consistently preferred keyboard to speech. Non-typists initially preferred speech for the MC editor only, but this effect was abolished by the end of the experiment. Structured interviews revealed a feeling on the part of subjects that "switching modality during a command was inherently disruptive."

Leggett & Williams (1984) conducted experiments in which 24 subjects entered and edited program code by speech and keyboard using a language-directed editor

(i.e. one having "knowledge of the underlying syntax"). The input and edit vocabularies contained 40 commands. The keyed edit commands could be abbreviated to the first three letters but, in the absence of information in their paper on this point, it must be assumed that input commands had to be keyed in full. They found that keyboard input was faster in that "subjects were able to complete more of the input and edit tasks by keyboard (70%) than by voice (50-55%)". However, speech had a much lower error rate on both the input task (3.8% versus 11.0%) and the edit task (1.2% versus 14.3%).

Damper, Lambourne & Guy (1985) compared the selection of 1-out-of-N simple commands by speech input and keypad in a television subtitling task: there were $N = 25$ distinct commands. Subjects concurrently entered literal subtitle text by keyboard. The 25 commands specified the position on screen of the subtitles, the background colour of the caption etc. They found that keypad entry of commands was 32% faster on average than ASR and had a much lower error rate. However, speech input reduced the time spent transferring between text and command entry. Overall, speech input increased the time to complete the subtitling task by 9%.

Martin (1989) considered two commonly-made assertions about the utility of speech input: (1) it is faster than typed input; (2) it provides an additional response channel over which workload can be spread, so increasing productivity in multi-modal interfaces. These assertions were investigated via a literature review (variously cited above) and an empirical study. The application studied was VLSI chip design using a highly-interactive, graphics-based package in which input tasks were of two types: drawing and command. Speech input was added to the existing (key-press) interface, and was intended "for entering verbally-oriented commands". It was compared to typed full-word input, single key-presses and mouse button clicks. Data were collected from 4 subjects. They were able to complete more tasks with the additional

speech channel – 62% versus 38%. In terms of average input times, there was no advantage for speech over mouse clicks, but mouse-entered commands were only a very restricted subset of the complete set. There was a 24% time advantage for speech over single key-presses, but this difference was only marginally statistically significant (t-test, $p < 0.1$). However, there was a clear and significant advantage (108%, $p < 0.001$) for speech over typed full-word command entry. As far as errors are concerned, Martin states:

> "In the case of speech input, the first-spoken instance of a command was assumed to be correctly-recognised, even if it was actually not ... because the focus is on the long-term utility of speech input ... once error rates approach 0%."

Hence, it seems that no sensible treatment of Martin's error data is possible. It is not clear how this aspect of the experimental design impacts on the other performance measures.

Karl *et al* (1993) compared speech selection of editing commands with mouse-activated selection in a word-processing task. Literal text entry was by keyboard and the mouse was used for direct manipulation in both cases. We consider this work to be relevant here because a mouse combines the pointing function with a rudimentary (1 out of 2 or 3) key-press selection mechanism, so implementing what amounts to a 'dynamic' keypad. 16 subjects achieved an average reduction in task time of 18.7% when using speech- rather than mouse-activation of commands. The authors apparently draw an implicit distinction between 'user' and 'system' errors in that they report "error rates due to subject mistakes were roughly the same" for speech and mouse, while quoting "recognition errors" of 6.3% for speech input but no error data for mouse entry. Again, this highlights another of the problems in

effecting a fair comparison of input media. What is the counterpart of a 'recognition error' in a key-press interface?

Overall, then, this review confirms the contradictory nature of the results commented upon by other workers. Notwithstanding some indications to the contrary from particular studies, the picture which emerges is largely pessimistic with regard to speech input – unless there are some special characteristics of the task. Multi-modal, highly-interactive applications do, however, appear good candidates for ASR, especially if hand occupancy is a feature.

Poock's Study: a Critical Analysis

A much more optimistic view of ASR than that which emerges from the above review was put forward by Poock (1980; 1982). His subjects entered (simulated) naval command and control instructions from a small-vocabulary on the ARPANET using a model T600 Threshold Technology isolated-word recogniser. They followed "a fixed scenario of instructions in which they accessed the ARPANET, logged on to different host computers, read messages, sent messages, checked for new mail, read files, and interconnected host computers". By "fixed scenario", Poock means that his subjects were given a specific task to perform, rather than having a pre-determined script to follow. Thus, they would be asked: *See if there is MAIL for EXPERIMENTAL.* A command vocabulary was suggested to the subjects but they were free to vary this. According to Poock (1980): "180 out of the possible 256 utterances ... were actually entered into the voice recognition unit although only about 75 utterances were actually needed in the experiment". The vocabulary was "entirely open with no branching to subsets of words". With one exception (a female civilian), the 24 subjects studied were male military officers. All 24 were

familiar with the functions of the system, and experienced in manual data entry to it, although none had previous ASR experience.

Subjects were allowed to practise with the recognition equipment until they felt "comfortable" with it (an average of 3.26 hours). They were divided into two groups according to their typing ability. These were further divided into two sub-groups: one half performed the speech entry first and the other performed the keyed entry first. The "fixed scenario" – designed to take about 10 minutes to perform – was repeated 4 times for each means of input, 8 times in all. Because the study involved use of a multi-user distributed computer system in which response times were non-deterministic, subjects were also given a secondary task to perform in idle time. This involved the transcription by hand of information from civilian aviation weather reports onto a data sheet. Actual elapsed times for the scenario ranged from 6 to 18 minutes.

Subjects also completed a questionnaire "concerning their opinions and views on manual typing input and voice input" some two weeks before the experiment and again at its completion. Broadly, they were positive in their views of speech input and this attitude was reinforced as a result of performing the experiment. For instance, the question "does voice input provide a better man-machine interface?" (asked only at the end of the experiment) elicited an average response of 5.80 on a 7-point scale – with 1 corresponding to *absolutely no* and 7 to *absolutely yes*.

As far as primary data entry is concerned, Poock's results contrast markedly with most of the studies reviewed in the previous section. In particular:

- speech was found to be 17.5% faster than keyboard entry, while ...
- typing led to 183.2% more errors.

Less controversially, in view of the remarks above concerning concurrent tasking, speech input also allowed subjects to transcribe 25% more weather information in the secondary task than was possible during manual entry.

The optimistic tone of Poock's study can be gauged from the quote:

> "In an era when so much is said and written about declining productivity in America, voice input technology may be one solution to helping reverse this trend."

In considering further the "conflicting results" reported in the literature, Simpson *et al* (1985) compare Poock's optimistic findings with those of McSorley (1981). In a computerised war-game scenario, McSorley concluded that manual entry was faster than speech input. The significant point here is that this work was performed in the same laboratory as Poock's and, apart from the specific task, "the majority of other factors (user group composition, training, equipment, and environment) were constant". No doubt it is facts such as this which lead Simpson *et al* to implicate "task-specific factors" as the primary source of conflict.

Are there, however, other possible contributory explanations? We have previously (Damper, 1988; Damper & Leedham, 1992; Damper, 1993) criticised Poock's experimental design for using unnecessarily verbose commands, and argued that this may have introduced a bias in favour of speech. Nye's (1982) finding that imposing a common command structure when comparing input media may unduly penalise one or other of them is apposite here. While commands had to be entered character-by-character when keyed, and terminated by pressing the return key, they were spoken as single (whole-phrase) utterances. For instance, subjects entered *SET ECHO* manually using 9 input actions (key-presses) but only a single action (utterance) in the speech condition. There seems no necessity whatsoever to require a user to

type *SET ECHO* in full when the natural form for keyboard input would be an acronym, word completion, or key assignment. It is also obvious that there will be approximately 3 times the probability of a keying error in entering *SET ECHO* rather than *SE*, assuming an equal probability of error in all keyings. By contrast, the recognition error under the same assumption will be constant, dependent upon the active vocabulary size.

In other words, Poock has failed to treat appropriately what Morrison *et al* (1984) call the transaction cycle (i.e. the number of required input actions for a system response). One approach might be to try to equalise the transaction cycle across the media being compared, on the grounds that this gives a fair comparison. Alternatively, and probably more reasonably, it might be sensible to minimise separately the transaction cycle for each medium, as this should ensure that each is used in a maximally effective fashion. In our view, Poock's failure to do either constitutes a methodological flaw in the experimental design, to the extent that we feel justified in calling it an error.

For the purposes of this paper, it is worth drawing a careful terminological distinction at this point. The word *command* is somewhat ambiguous in that it could either refer to the sequence of input actions effected by a user, or to the symbol string which that sequence of actions accesses. Since we are centrally concerned with the mapping between these two, we avoid ambiguity by reserving *command* to mean the sequence of input actions. Thus, it is meaningful to refer to "verbose" and "terse" commands even when entry of these has an identical effect in terms of the system response. When we wish to evoke the latter sense, we use the term *command-string* (synonymous to full-word command).

In view of the above discussion of Poock's experimental design, it was decided in

this work to base our comparison of speech and keying on the use of terser (and arguably more reasonable) commands for at least the keying condition.

Experiments

The hypothesis under test is that Poock's experimental design, by the use of unreasonably verbose commands in the keying condition, introduced a bias in favour of the speech condition. Ideally, to test this, we would repeat his study in all particulars changing only the detailed input actions and their mappings to command-strings. However, it was clearly not possible for us to use the same subjects and (classified) equipment, so that some differences are inevitable. Accordingly, we have not tried to replicate the earlier experiment exactly and there are several potentially important differences. However, as argued in the Discussion below, we believe that the *principal* difference between this study and Poock's is the use of terser commands for (at least) the keying condition. Where there were differences, we attempted as far as possible (within the constraints of resources available to us) to make these such as to favour the speech condition, so as to counter our working hypothesis and provide a maximally stringent test of it.

*** TABLE 1 ABOUT HERE ***

Given that keyed commands are to be abbreviated, it seemed worth comparing these with *both* 'abbreviated' and full-word (whole-phrase) spoken commands. In place of Poock's single speech/keying comparison, therefore, we have two comparisons. Thus, the overall study was divided into Experiment 1 and Experiment 2 (denoted *E1* and *E2* respectively), in which only the spoken commands differed. In all cases, subjects entered keyed commands as acronyms, as shown in Table 1. For example,

REPEAT was keyed as *R* and GO TO ECHO as *GTE*, terminated by activation of the return key. (The significance of the upper case here is purely notational: to distinguish keyed from spoken commands. All keyboard entry was case-insensitive.) The two speech/keying comparisons were as follows:

- In Experiment 1, spoken commands were entered as acronyms, e.g. subjects said *gte* in the case of GO TO ECHO. (Some acronyms were also lexical words, e.g. *spa*: subjects were allowed to speak these in whichever way was most natural.) Because of the difficulty of recognising single letter-names reliably, however, single-word command-strings such as REPEAT were exceptionally entered as spoken whole words. This was done so as not to disadvantage speech input unduly relative to keyed input.

- In Experiment 2, spoken commands were entered as complete phrases, i.e. subjects said *repeat* and *go_to_echo*.

In summary, Experiment 1 compares spoken and keyed command acronyms (with the exception that single-word commands were spoken as whole words) while Experiment 2 compares the 'natural' command language for each medium, i.e. acronyms for keyed input and complete phrases for spoken input.

Since this doubles the number of experimental treatments, we used 12 young adult subjects in place of Poock's 24. 11 of the 12 were undergraduate students, 3 were female, and 6 were from a non-technical background. None had previously used speech recognition equipment. Keyboard experience was variable from virtually none among some of the non-technical students to considerable: this should favour the speech condition relative to Poock's study since all his subjects were experienced in manual entry to the experimental system. Because of the time availability of our

subjects (restricted to university lecture slots), each of Experiments 1 and 2 was designed to fit into a one-hour session. Hence, Poock's "fixed scenario" was simplified: we used 28 distinct commands as listed in Table 1 in place of his "about 75". Again, this should favour speech over keying, as recogniser error rate is expected to increase with vocabulary size. Moreover, we eliminated the numeral 5 from the vocabulary because of its well-known confusability with 9. A SYS300 200-word recogniser (Interstate Voice Products, 1984) was used, broadly comparable in function and performance to the Threshold Technology T600, hosted by an Amstrad PC1512. An initial quarter-hour session (considerably less that the 3.26 hours for Poock's subjects) was allocated to familiarisation with the speech recogniser and the task.

The recogniser was trained at the beginning of each one-hour experimental session. Because we were not using a distributed computer system, no secondary task was allocated for subjects to perform in idle time. (In principle, this could be a potentially important difference since the advantages of speech are likely to come to the fore in situations of concurrent tasking.) Since our subjects were not working with a real command and control system of which they had prior working experience, they were prompted for data entry by a pre-determined script in place of Poock's "fixed scenario". Prompts appeared on-screen, one at a time, in the form of the full-word command on one line (as in the first column of Table 1) and the corresponding acronym on the next line (as in the second column of this table). These were considered to be adequate for the subjects to infer the required input action(s).

Table 2 shows the script entered: for brevity, this is specified in the form of acronyms (rather than in the form of prompts issued to the subjects). The script was entered 16 times in all (4 times each for speech and for keying, in each of the sessions allocated to Experiments 1 and 2). The script was identical in all cases, involving 28 distinct commands and 79 in total, except that the form of the on-screen prompts

sometimes differed as described above.

*** TABLE 2 ABOUT HERE ***

The script was designed to take about 2 minutes to complete (i.e. 16 minutes per session) so that, allowing rest time and some 15 minutes to train the recogniser, individuals could complete their work within the available hour. Subjects were instructed to adopt a 'first-final' strategy, i.e. not to attempt to correct any errors which arose but to ignore them. In order not to disadvantage ASR relative to keying, all experiments were conducted in a sound-proof, quiet room. Subjects were not observed, so that no distinction was (or could be) drawn between 'user' and 'system' errors.

At the outset, subjects were allocated to one of 4 equal-size groups (A, B, C or D) of 3 according to the order in which they performed Experiments 1 and 2 and, further, the order in which they did the speech and keying halves of the experiments. Table 3 depicts the overall experimental design. In this table, *E1* and *E2* represent Experiments 1 and 2 as before, while s and k denote the speech and keying conditions, respectively. We will refer to each of the identified halves of an experiment (*E1s, E1k*, etc.) as a *block*. In each block, therefore, there are 4 repetitions of the data entry from the script. We will refer to these repetitions as *Runs* 1 to 4, irrespective of the specific block. To avoid confusion between the two (similar) command sets used for Experiments 1 and 2, subjects were given a break of at least two days between Sessions 1 and 2 (i.e. between Blocks 2 and 3).

*** TABLE 3 ABOUT HERE ***

As in Poock's work, subjects completed a questionnaire probing their attitude to speech input before and after the experiments. However, our subjects received a

similar questionnaire immediately before Session 1 (rather than about two weeks before).

Results

A particular subject proved to be a very problematic user of speech input. His errors and timings were some 100% higher than the average. Accordingly, in line with our comments above about a maximally stringent test of our working hypothesis, data for this subject were removed in order to present speech input in as fair a light as possible. The literature reveals that this is not an uncommon necessity. For instance, Damper *et al* (1985) state: "One of the subjects suffered from very bad performance, due partly to an inability to adjust to the consistent pronunciation required". Similarly, Martin (1989; p. 366) writes: "One subject encountered *[serious]* difficulties in this regard, because he frequently talked to himself."

Table 4 shows the results obtained from each of *E1s*, *E1k*, *E2s* and *E2k*, averaged over all (11) subjects and runs. None of the observed differences between Experiments 1 and 2 are significant. For instance, average script entry time for *E1s* (spoken acronyms) was 129.6 seconds, while for *E2s* (complete spoken phrases) it was 133.8 seconds (t-test, $p > 0.9$). The average numbers of speech errors were 6.8 and 6.9 respectively, almost identical. The average numbers of keying errors were 1.7 and 1.25 (t-test, $p \sim 0.4$). For this reason, data from the two experiments were pooled for subsequent analyses.

*** TABLE 4 ABOUT HERE ***

Figure 1 shows input time averaged across all blocks for the 11 subjects under speech and keying conditions, as a function of the run number. It is apparent

from the figure that keying is consistently faster than speech, but only slightly so. As in Poock's work (1982; Figure 2), there is a trend for reducing input time with experience in performing the task, no doubt resulting from increased familiarity with the command vocabulary, the script and the requirements of the task. However, this trend is less marked in our data, almost certainly because our subjects (as a result of on-screen prompting) did not have to decide for themselves what commands to use – so reducing any advantage gained through familiarity. (Also, our subjects repeated the data entry from the script 8 times for each condition, c.f. 4 times for Poock's "fixed scenario"). Input time is asymptotic to about 113 seconds for keying, and about 129 seconds for speech.

*** FIGURE 1 ABOUT HERE ***

Figure 2 shows the average numbers of errors for the 11 subjects under speech and keying conditions as a function of run number. It is readily apparent from the figure that speech is enormously more error-prone than keying. There does not seem to be any systematic variation of error rate with run number for either input medium. This is in contrast to Poock's data (1982; Figure 3) which shows a fairly consistent reduction of speech errors across trials, and an initial sharp fall for keying followed by a small increase.

*** FIGURE 2 ABOUT HERE ***

Table 5 shows average input times and numbers of errors pooled across all 11 subjects, both experiments and all 4 runs. It is seen that speech was 10.6% slower than keying, although this difference was not statistically significant (t-test, $p \sim 0.87$). Speech was enormously more error-prone, however, leading to a factor of 4.6 more

errors (i.e. a 360.4% increase) relative to keying. This difference was highly significant (t-test, $p < 0.001$).

*** TABLE 5 ABOUT HERE ***

Two points of interest emerged from the questionnaires. First (although this aspect of speech input was not actually investigated here), all subjects believed that ASR would allow greater freedom to perform other tasks. This was true for questionnaires completed both before and after the experiments. Second, before performing the experiments only 14% of subjects felt that speech input would be more frustrating than keying. Afterwards, this figure rose to 57%. Clearly, the subjects had expectations of ASR which were not fulfilled in practice, at least in this study. One possible explanation for this might be the effectively obsolescent recogniser employed, for reasons of parity with Poock's experimental set-up. His subjects were making their judgements in the context of the state of technology *circa* 1980 whereas ours (only half of whom were from a technical background, however) may have had higher expectations as a result of more than a decade of rapid advances.

Discussion

In our experiments, we have tried to effect a fair comparison of speech and keying by minimising the transaction cycle (Morrison *et al*, 1984) for the two media. That is, we have varied the mappings from commands (sequences of input actions) to command-strings elicited to suit the properties of the individual media. We find that isolated-word speech input in a simulated command and control application is slightly slower than keying (10.6%) although not significantly so. However, ASR is enormously more error-prone (360.4%). This is in stark contrast to the similar work

of Poock (1980; 1982) who claimed speech was 17.5% faster while keying produced 183.2% more errors. The major differences between our work and Poock's are:

1. the use of shorter commands in the case of keying;
2. the absence of concurrent tasking;
3. the use of a (prompted) script in place of a "fixed scenario";
4. significantly reduced familiarisation time.

By implication, the disparity between the two sets of results can be attributed to some combination of these factors.

Some light can be shed on the relative impact of these 4 differences by a simple comparison of keystrokes for the entry of acronyms and full-word keying. 198 keystrokes were necessary for the entry of the script using keyed acronyms (Table 2). From the keystroke savings listed in Table 1, the required number for full-word keying – as in Poock's study – would have been 507. Hence, assuming both input time and number of errors to be simply proportional to the number of keystrokes, we would expect the average input time and number of errors to be 305.0 ms and 3.82 respectively (c.f. 119.1 ms and 1.49 in Table 5). Accordingly, we would expect speech to be 56.8% faster than keying when commands are entered in full while typing would still have 41.1% fewer errors than speech. Not unexpectedly, ASR now appears in much more favourable light, with speech input faster than typing and the error rate much closer to that for keying.

We interpret these figures as broad support for the notion that verbose commands do indeed disadvantage keying relative to speech. However, our estimates do not align perfectly with Poock's findings in that keying remains superior to speech in

terms of error rate. One strong possibility is that keying errors increase disproportionately with command length, so that the basic assumption underlying the above extrapolation is violated. Apart from this, the most likely causes of the remaining discrepancy are differences 2 and 3. As a result of both of these, Poock's subjects had a significantly higher cognitive load imposed during the task than did our subjects, and it is well accepted that speech confers an advantage in such situations. Difference 4 appears unimportant in view of the fact that Figures 1 and 2 offer very weak evidence of any differential trend towards improved ASR performance over keying performance with experience of the task.

Hence, it is our belief at this stage that the *important* difference between our study and Poock's is our use of terser commands in the keying condition. Thus, Poock's optimistic claims for the superiority of speech are mainly – although probably not entirely – an artifact of an ill-advised choice of keying commands. Future work is planned in which we will add concurrent tasking, thereby increasing the subjects' cognitive loading, to our experiments to confirm this point. We also intend explicitly to add full-word entry of keyed commands as a direct test of our hypothesis. (This was not done in this study because of our concern during the initial experimental design phase to limit the number of treatments, so that too much would not be expected of our subjects.)

It could be argued that the work described here is dated, based as it is on *circa*-1980 ASR technology, and is no longer relevant. We would refute this on three counts. First, as outlined in our Introduction, early studies of the usability of speech recognition rightly achieve status as key pieces of literature. It is important that the claims of earlier workers remain open to critical re-appraisal, especially if they are to be cited as justification for employing ASR widely in real, present-day applications. Second, in spite of the emergence of large-vocabulary speech recognisers in

the last few years, most practical applications remain small-vocabulary in nature, and this is likely to be the case for some time to come. Our findings are relevant to these present and future applications. We concede, of course, that today's large-vocabulary recognisers should be well capable of reducing the error rate in small-vocabulary applications very significantly, relative to obsolescent recognisers like the T600 and SYS300. However, a degree of (costly) re-engineering of the language models would be necessary, plus these systems are currently too expensive to be used in this way. Third, we believe that this work contributes to the development of a methodology for comparing speech with competitor input media by demonstrating that specific differences in the interface (here, the mapping between commands – i.e. input actions – and the command-string accessed) in the two cases can profoundly affect results.

This demonstration has important implications for studies of the usability of ASR, irrespective of the technical capabilities of the particular recogniser used. The human factors of speech input is a difficult area of study: we believe that the work reported here contributes to its development. Without such development, we are unlikely ever fully to exploit emergent speech technology. What is required is the "creation of a speech paradigm, analogous to the window and icon paradigm" (Sharman, 1993), currently so ubiquitous in human-computer interaction, and founded on a proper understanding of the part that speech input can best play in the multi-modal interfaces of the future.

Conclusions

We have described two experiments comparing speech and keying input in a simulated naval command and control task. In both experiments, terse commands based

on acronyms are used for the keying interface. The spoken commands differ, however, being (essentially) spoken acronyms in one experiment and complete phrases in the other. The results, in terms of input speed and numbers of errors encountered, do not vary significantly between the two experiments and so data from the two were pooled for further analysis.

The experiments were modelled on the earlier work of Poock (1980; 1982) who claimed that speech input was faster and far less error-prone than keying. However, our findings are very different: speech is shown to be slower (although not significantly so) and enormously more error-prone. Extrapolation from our data – based on a simple analysis of keystrokes – to the situation where keyed commands are entered in full suggests that this difference is mainly (but not entirely) attributable to the use by Poock of unnecessarily verbose commands which are a poor fit to the requirements of a key-press interface. It is argued that this constitutes a flaw in Poock's experimental design, which is corrected in our study by attempting to minimise the transaction cycle for keying as well as for speech. It is also likely that the absence of concurrent tasking (or any necessity for subjects to decide for themselves what commands to enter) in our new experiments played a part in the observed differences. We intend to confirm this interpretation by adding secondary tasking and full-word command entry in future work.

This study contributes to the development of a methodology for comparing speech with a competitor input medium by demonstrating that specific but subtle differences in the interface can profoundly affect results. A fair comparison of media requires that each is used along with an interface design tailored to the particular capabilities and requirements of that medium. One reasonable way to achieve this is to attempt to minimise the transaction cycle (Morrison *et al*, 1984) – the number of input actions necessary to elicit a system response – in each case.

Acknowledgements

The authors are grateful to Richard Sharman, Speech Consultant, IBM (UK) Laboratories, Hursley, who suggested the keystroke analysis, and to Sue Lewis, Faculty of Mathematical Studies, University of Southampton, who made valuable comments on the experimental design. Thanks are also due to our subjects who gave their time entirely freely.

References

AINSWORTH, W.A. (1988) *Speech Recognition by Machine*, Peter Peregrinus, London.

AMALBERTI, R., CARBONELL, N. & FALZON, P. (1993) "User representations of computer systems in human-computer speech interaction", *International Journal of Man-Machine Studies*, **38**, 547–566.

CHAPANIS, A. (1975) "Interactive human communication", *Scientific American*, **232**, 36–42.

CHAPANIS, A., PARRISH, R.N., OCHSMAN, R.B. & WEEKS, G.D. (1977) "Studies in interactive communication: II. The effects of four communication modes on the linguistic performance of teams during cooperative problem solving", *Human Factors*, **19**, 101–126.

DAMPER, R.I. (1988) "Practical experiences with speech data entry", in *Contemporary Ergonomics 1988*, E.D. McGaw (ed.), Taylor and Francis, London, pp. 92–97.

DAMPER, R.I. (1993) "Speech as an interface medium: how can it best be used?", in *Interactive Speech Technology: Human Factors Issues in the Application of Speech Input/Output to Computers*, C. Baber and J.M. Noyes (eds.), Taylor and Francis, London, pp. 59–71.

DAMPER, R.I., LAMBOURNE, A.D. & GUY, D.P. (1985) "Speech input as an adjunct to keyboard entry in television subtitling", in *Human-Computer Interaction – INTERACT '84*, B. Shackel (ed.), Elsevier (North-Holland), pp. 203–208.

DAMPER, R.I. & LEEDHAM, C.G. (1992) "Human factors", in *Speech Processing*, C.G. Rowden (ed.), McGraw-Hill, Maidenhead, pp. 360–393.

INTERSTATE VOICE PRODUCTS (1984) *Voice Recognition System Model SYS300: Operation and Maintenance Manual*, TMP00702199-2, 1849 W. Sequoia Avenue, Orange, CA 92668.

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.

LEA, W.A. (1980) "The value of speech recognition systems", in *Trends in Speech Recognition*, W.A. Lea (ed.), Prentice-Hall, Englewood Cliffs, NJ, pp. 3–18.

LEE, K-F. (1989) *Automatic Speech Recognition: the Development of the SPHINX System*, Kluwer, Dordrecht.

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.

MARTIN, G.L. (1989) "The utility of speech input in user-computer interfaces", *International Journal of Man-Machine Studies*, **30**, 355–375.

MCSORLEY, W.J. (1981) *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 command structure", *International Journal of Man-Machine Studies*, **21**, 49–63.

NYE, J.M. (1982) "Human factor analysis of speech recognition systems", *Speech Technology*, **1**, 50–57.

POOCK, G.K. (1980) "Experiments with voice input for command and control: using voice input to operate a distributed computer network", *Naval Postgraduate School Report, NPS55-80-016*, Monterey, CA.

POOCK, G.K. (1982) "Voice recognition boosts command terminal throughput", *Speech Technology*, **1**, 36–39.

SCHURICK, J.M., WILLIGES, B.H. & MAYNARD, J.F. (1985) "User feedback requirements with automatic speech recognition", *Ergonomics*, **28**, 1543–1555.

SHARMAN, R.A. (1993) "Speech interfaces for computer systems: problems and potential", *Displays*, **14**, 21–31.

SIMPSON, C.A., MCCAULEY, M.E., ROLAND, E.F., RUTH, J.C. & WILLIGES, B.H. (1985) "System design considerations for speech recognition and generation", *Human Factors*, **27**, 115–141.

WELCH, J.R. (1977) "Automated data entry analysis", *Rome Air Development Center Report RADC TR-77-306*, Griffiss Air Force Base, NY.

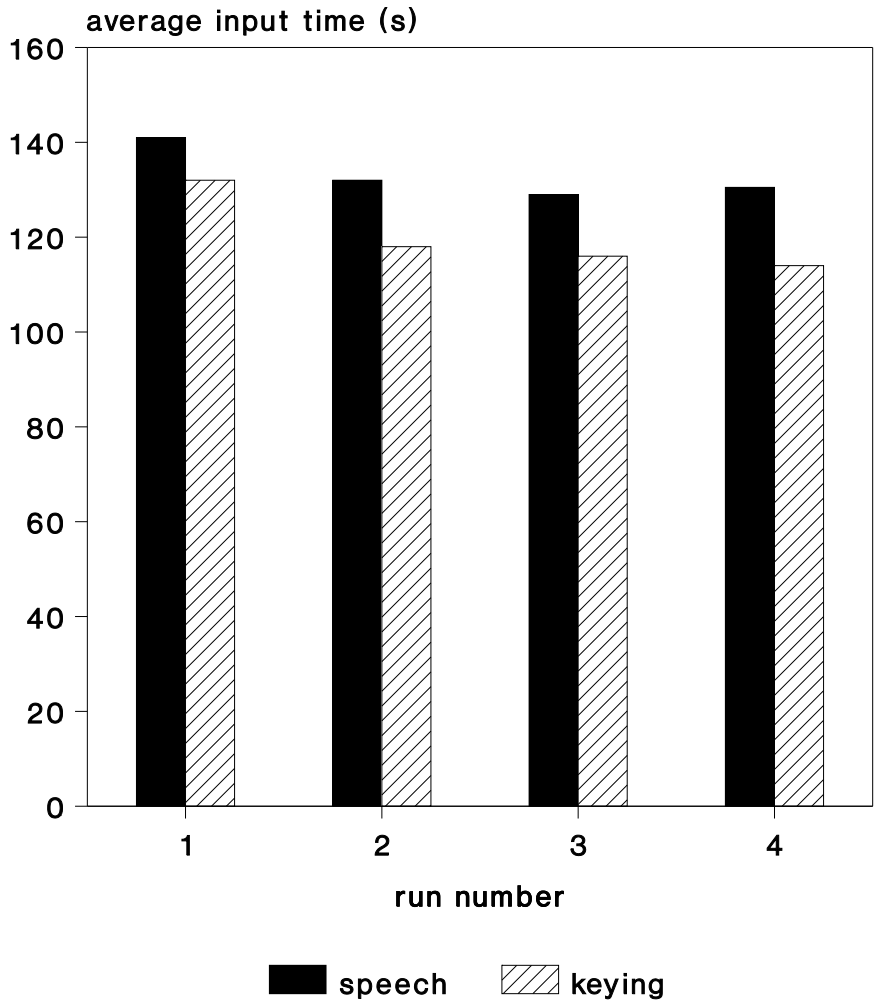

Figure 1: Average input times for speech and keying as a function of run number: pooled data from Experiments 1 and 2 for 11 subjects.

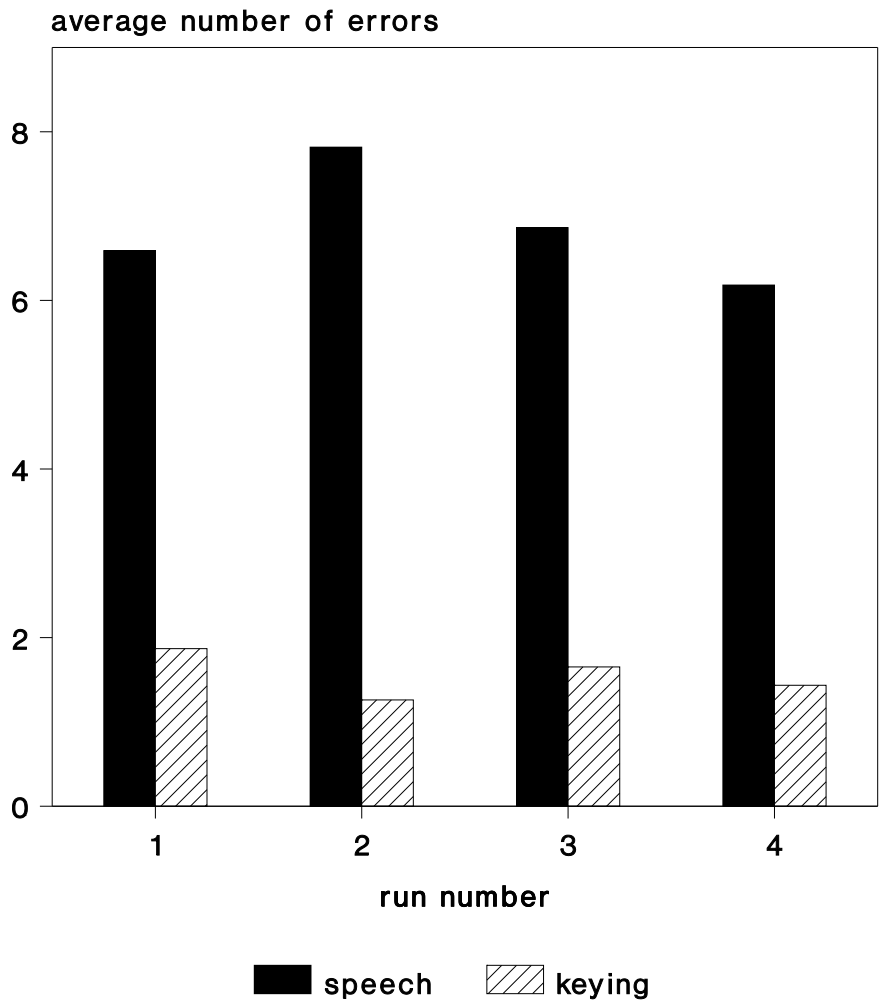

Figure 2: Average number of input errors for speech and keying as a function of run number: pooled data from Experiments 1 and 2 for 11 subjects.

Full Command ('Command String')	Keyed Command	Keystroke Saving	Spoken Command (*E1*)	Spoken Command (*E2*)
GET MESSAGE	*GM*	9	*gm*	*get_message*
GO TO WAYPOINT	*GTW*	11	*gtw*	*go_to_waypoint*
LAUNCH RESCUE CRAFT	*LRC*	16	*lrc*	*launch_rescue_craft*
LOCATE TRACE	*LT*	10	*lt*	*locate_trace*
LOCK RADIO	*LR*	8	*lr*	*lock_radio*
MILES	*M*	4	*miles*	*miles*
MONITOR RESCUE CRAFT	*MRC*	17	*mrc*	*monitor_rescue_craft*
RADIO BASE	*RB*	8	*rb*	*radio_base*
RECEIVE RESCUE CRAFT	*RRC*	17	*rrc*	*receive_rescue_craft*
REPLY	*R*	4	*reply*	*reply*
RESPOND TO RADAR	*RTR*	13	*rtr*	*respond_to_radar*
SEND COORDINATES	*SC*	14	*sc*	*send_coordinates*
SEND MISSION STATS	*SMS*	15	*sms*	*send_mission_stats*
SET AUTO PILOT	*SAP*	11	*sap*	*set_auto_pilot*
SET PATROL AREA	*SPA*	12	*spa*	*set_patrol_area*
SET RADAR LOCAL	*SRL*	12	*srl*	*set_radar_local*
SET RADIO TO SWEEP	*SRS*	15	*srs*	*set_radio_to_sweep*
SET SWEEP	*SS*	7	*ss*	*set_sweep*
SET WAYPOINT	*SW*	10	*sw*	*set_waypoint*
SLOW AHEAD	*SA*	8	*sa*	*slow_ahead*
0	0	0	*zero* or *oh*	*zero* or *oh*
1	1	0	*one*	*one*
2	2	0	*two*	*two*
3	3	0	*three*	*three*
4	4	0	*four*	*four*
7	7	0	*seven*	*seven*
9	9	0	*nine*	*nine*
,	,	0	*comma*	*comma*

Table 1: Command vocabulary employed in this study. The keystroke saving resulting from use of acronyms rather than full commands is also listed for each command. Subjects were prompted with the full command (first column) and the acronym corresponding to the keyed command (second column).

SRS SW 1 0 3 , 2 7 4 SAP LR GM R SW 3 9 0 , 4 3 7 GTW SRL SS 4 M SPA 3 9 1 , 4 3 7 LR SA SPA 3 9 2 , 4 3 7 SA RTR LT SW 3 9 2 , 4 3 7 GTW RB SC 3 9 2 , 4 3 7 LRC MRC RRC RB SMS SW 1 0 3 , 2 7 4 SAP

Table 2: Experimental script shown, for brevity, in the form of acronyms. The script involves 79 commands in total, of which 28 were distinct. 198 keystrokes were required for keyed entry.

Session	Block	A	B	C	D
1	1	*E1s*	*E1k*	*E2s*	*E2k*
1	2	*E1k*	*E1s*	*E2k*	*E2s*
2	3	*E2s*	*E2k*	*E1s*	*E1k*
2	4	*E2k*	*E2s*	*E1k*	*E1s*

Table 3: Experimental design for the comparison of speech (s) and keying (k) in Experiments 1 and 2.

measure (average)	*E1*	*E2*
time (s) – speech	129.6	133.8
time (s) – keying	122.6	115.6
errors – speech	6.8	6.9
errors – keying	1.7	1.3

Table 4: Input times and number of errors for speech and keying averaged across all subjects and runs for Experiments 1 and 2.

	speech	keying	% difference
time (s)	131.7	119.1	10.6
errors	6.86	1.49	360.4

Table 5: Average input times and errors pooled across all subjects, experiments and runs.