Int. J. Human-Computer Studies (2001) **54**, 637–662
doi:10.1006/ijhc.2000.0456
Available online at http://www.idealibrary.com on IDEAL®

Designing habitable dialogues for speech-based interaction with computers

K. S. Hone

Department of Information Systems and Computing, Brunel University, Uxbridge Middlesex UB8 3PH, UK. email: kate.hone@brunel.ac.uk

C. Baber

School of Electronic & Electrical Engineering, University of Birmingham, Birmingham B15 2TT, UK. email: c.baber@bham.ac.uk

(Received 13 October 1999, accepted in revised form 31 December 2000, and published electronically 7 February 2001)

Habitability refers to the match between the language people employ when using a computer system and the language that the system can accept. In this paper, the concept of "habitability" is explored in relation to the design of dialogues for speech-based systems. Two studies investigating the role of habitability in speech systems for banking applications are reported. The first study employed a speech-driven automated teller machine (ATM), using a visual display to indicate available vocabulary. Users made several distinct types of error with this system, indicating that habitability in speech systems cannot be achieved simply by displaying the input language. The second study employed a speech input/speech output home banking application, in which system constraints were indicated by either a spoken menu of words or a "query-style" prompt (e.g. "what service do you require?"). Between-subjects comparisons of these two conditions confirmed that the "menu-style" dialogue was rated as more habitable than the "query-style". It also led to fewer errors, and was rated as easier to use, suggesting that habitability is a key issue in speech system usability. Comparison with the results of the first study suggests that for speech input, spoken menu prompts may be more habitable than similar menus shown on a visual display. The implications of these results to system design are discussed, and some initial dialogue design recommendations are presented.

1. Introduction

In Human Factors research, "compatibility" describes the relationship between the operation of an artefact and a person's expectations as to how that artefact should be operated (Sanders & McCormick, 1992). For example, the operation of a control knob can be influenced by the "clockwise-to-increase" stereotype such that one expects a clockwise rotation of the control knob to lead to an increase in a controlled variable, e.g. volume on a radio. These expectations may be influenced by the physical appearance of the artefact, the person's previous experience of similar artefacts, or by cultural stereotypes concerning how classes of artefacts are used.

In this paper, we will explore a concept which is closely linked to compatibility: habitability. The concept of "habitability" was introduced by Watt (1968) to describe the

1071-5819/01/040637 + 26 $35.00/0

match between the language people choose to employ when using a computer system and that which the system can accept. Watt (1968) defines a habitable computer language as "one in which its users can express themselves without straying over the language's boundaries into unallowed sentences". Watt (1968) was concerned with the development of question–answer systems for specialized dialogue domains, such as flight timetable enquiries. In such domains, users type natural language queries, e.g. "I want to get to Heathrow by noon in order to catch the earliest flight to Lisbon with dinner en route: which flight shall I take?" Watt (1968) argued that a challenge for such systems is that semantically valid questions can be phrased in a wide variety of different ways. A *minimally habitable* system will accept one paraphrase sentence for each meaning the user might want to express (Watt calls this a "semantically adequate" system). In contrast, a *fully habitable* system requires that every possible paraphrase be acceptable. Given the inevitable combinatorial explosion to which this would lead, Watt (1968) argues that real systems can only ever approximate full habitability. However, habitability can be optimized by allowing the system to accept multiple paraphrases and ensuring that these match as closely as possible to the sentences which users will actually produce in real interactions. This raises two significant human factors issues: (1) how can one define an adequate set of paraphrases which people will actually use?; (2) how can one inform users of the set of paraphrases from which they are free to choose? The first question implies that users will have unlimited opportunity to say what they feel (which raise the spectra of combinatorial explosion), while the second implies that people will be constrained in their choice of utterance (which raises the issue of how constraint can be handled). Ogden (1988) notes that departures from full habitability effectively represent constraints on what the user can input to the system. Users must operate within these constraints if they are to interact effectively with the computer system, as any input falling outside the constraints of the system will fail.

When Watt (1968) coined the term habitability, he referred to the match between sentences which users produce and those that the system can accept. More recently, similar points have been made by Furnas, Landauer, Gomez and Dumais (1987) regarding the individual vocabulary items that people use. They introduce the "vocabulary problem", which is that new or intermittent users of computer systems often use the wrong words for interface objects or actions. Their empirical research on this issue highlighted great variability in spontaneous word choice, leading them to suggest that the approach of using a single vocabulary item for each system object or action† will result in 80–90% failure rates. Furnas *et al.* (1987) conclude that "many, many alternative access words are needed for users to get what they want from large and complex systems" (p. 971); a statement which clearly mirrors Watt's proposals that systems should be designed to accept multiple paraphrases; but which also leads to problems of combinatorial explosion, i.e. how many alternatives constitute "many, many", and how does one ensure all necessary alternatives are included?

Habitability can thus be considered at several different levels. Watt's (1968) original concern was with what might be termed "syntactic habitability", referring to the match between the syntax the system can accept and that which the user produces. Also implicit in Watt's writing is the concept of "semantic habitability", referring to the match between

†Note that in Watt's terms such a system might be classed as "semantically adequate".

the meanings which users want to express (given the domain) and the meanings that can be accepted by the system. In addition to syntactic and semantic habitability, the Furnas *et al.* (1987) research on system vocabulary addresses what might be called "lexical (or word level) habitability" (see Ogden, 1988).

2. Constraints on speech-based interaction with computers

Interest in habitability as an issue has waned over the last decade as graphical user interfaces (GUIs) have come to dominate in human–computer interaction. The emphasis on direct manipulation within GUI design has removed many habitability problems because users are constrained to interact only with objects which are visible within the interface (see Shneiderman, 1998). Speech input, however, demands that the problems of habitability, as defined by authors such as Watt (1968), are revisited. Not only does speech potentially suffer from the kind of problems that they discussed, but it also potentially suffers from additional habitability problems because appropriate user inputs may fail during the speech recognition process (i.e. there may be a misrecognition). The notion of constraint is also highly relevant to speech systems, as designers frequently use lexical and syntactic constraints in an attempt to improve recognition accuracy (Hone & Baber, 1999).

In this section, we define speech system habitability in terms of the different levels of constraint which can operate over user utterances. In addition to the semantic, syntactic and lexical levels defined above, we have found it necessary to introduce two additional constraint levels in order to fully describe user interactions with speech technology. The first concerns the manner in which "legal" utterances change depending on the local dialogue context. We have termed this "dialogue level constraint". The second concerns variability in recognition accuracy, which we have termed "recognition constraint". The resulting five levels of constraint in speech-based interactions are summarized in Table 1. The levels are ordered according to the scope of the constraints. The semantic constraints have the widest scope, as they apply at the level of the interaction as a whole; whereas the recognition constraints have the narrowest scope, applying at the level of word or sub-word units.

Each of the levels of constraint shown in Table 1 will be introduced in more detail in the sections which follow. The discussion aims to highlight the relationship between each level of constraint and "acceptable" user behaviour. It also aims to illustrate some ways in which speech technology introduces constraint and explain why this helps speech processing. It is important to note that system design will always introduce constraints on what utterances can be accepted from the user, be these the explicit intention of the system designer, or an unplanned consequence of the various design decisions taken.

At this point it may be helpful to highlight the difference between the way the term "constraint" is used in this paper and the way constraint is often discussed in natural language understanding literature (as exemplified by Allen, 1995). In the current paper, the emphasis is upon the constraints which machines impose on user behaviour. In natural language understanding research the emphasis is often upon how "natural" constraints in the way people use language can aid the processing of that language. In fact, these perspectives should be linked, as the way users will actually constrain their

TABLE 1
Constraints and their influence on speech-based interaction

Constraint type	Level	Effect on user
Semantic	Entire interaction	Restriction on services/functions available from the system
Dialogue	Between pairs of utterances (or larger units)	Dynamic changes in meaningful utterances, dependent on context within interaction
Syntactic	Within utterance (sentence or phrase)	Restriction on order in which words can be spoken, allowable sentences
Lexical	Single words (or simple phrase)	Which words or simple phrases can be spoken
Recognition	Single phonemes, or words/phrases	Manner in which words are spoken

language when interacting with a computer will relate to the constraints which they believe are operating within that context.

2.1. SEMANTIC CONSTRAINT

Semantic constraint governs the meanings which users are permitted to express in their inputs to a system. Typically, the application domain will restrict the scope of meaningful inputs. For instance, in a home banking system users will expect to be able to refer to objects such as bank accounts and to actions such as money transfers. They may not expect to be able to refer to objects and services which they believe to be outside of the scope of banking services (e.g. ordering a pizza). It is this restriction which makes it feasible to develop spoken language systems with existing technology.

Achieving habitability at the semantic level relies on users and designers having mutual understanding of the boundaries of the application domain. Problems arise when users try to refer to objects which are unavailable or try to perform illegal actions on interface objects. Alternatively, users may not be aware of all the services available from a system and will therefore avoid expressing meanings which are actually acceptable.

In this paper, we assume that the imposition of semantic constraint in working speech systems is a straightforward matter of deciding what services to offer. While this is a considerable simplification, it is adequate for our purposes, especially given the difficulties of examining semantic constraint in a controlled experimental setting. Experiments often depend upon the user being told what tasks they should complete with the system, thus removing the potential to find what tasks they might expect to complete with such a system in the real world [see Ogden (1988) for further discussion of this issue].

2.2. DIALOGUE CONSTRAINT

Dialogue constraint is defined as constraint over the meanings that users are permitted to express given the local dialogue context. Many speech-based interactions are made up of adjacency pairs, for example where a question is followed by an answer (Bunt, Leopold, Muller & van Katwijk, 1978; Waterworth, 1982; Baber, 1993). The expectation is that where a person is asked a question, the possible response set will be constrained to include only valid answers to that question. For example, if someone is asked a question such as "When should the cash transfer be made?", the meaning of their reply should refer only to a point in time (e.g. "tomorrow"; "15:00 on 1/5/2000"). We present this level as below semantic constraint, as while it also relates to permitted meanings, its scope in narrower, extending only to the local context.

In the design of speech systems, dialogue constraints are sometimes introduced through the use of restricted response sets that make sense given the local dialogue context. For example, if the computer has just used the prompt, "Which home banking service would you like?", the designer might reduce the acceptable response set to those which refer to a home banking service. An alternative programming approach is to use such questions to generate expectations about what the user will say and use this to aid subsequent interpretations of the user input (e.g. Smith, 1998). This is a potentially less prescriptive approach to dialogue constraint, allowing the user to change topics if they desire (i.e. producing mixed initiative dialogue rather than computer-controlled dialogue). The key question for dialogue habitability is the extent to which users will interpret computer questions as cues to restrict their utterances.

2.3. SYNTACTIC CONSTRAINT

Syntactic constraint has already been defined in terms of the number of different paraphrases which are allowed for expressing a given meaning. More formally, syntactic constraints are usually defined in terms of a set of rules, or grammar, which specifies the form of acceptable utterances. For instance, a very simple grammar may state that an object (e.g. a bank account name) followed by a service name (e.g. balance) is an acceptable input format. This grammar would accept the utterances "current account balance" and "savings account transfer" but not the utterances "balance savings account" or "my balance please".

In speech recognition systems there are various ways in which syntactic constraints can be employed. At the strictest level, the system may be programmed to accept only specific sentences. An alternative is that syntactic rules are employed dynamically during the recognition process, for example, if the system recognizes "what is the balance of my", then the recognition set for the next input word might be reduced to include only types of bank account (e.g. savings or current account). The process can be made more flexible by using keyword-spotting techniques to pick out only the key parts of such sentences (for example, "balance" and "savings account"). Simple rules, based for example on word order, may be used to interpret sentence meaning when such techniques are used. However, this approach is limited. Consider, for example, the two sentences, "transfer £100 from my current account to my savings account" and "transfer £100 to my current account from my savings account". In order to correctly interpret these sentences the system needs more than rules based on keyword order, e.g. it also needs to understand

the meaning of "to" and "from" in these sentences. Alternative approaches include the "conversation-for-action" paradigm used by Wolf, Kassler, Zadrozny and Opyrchal (1997) in their Conversational Machine. This employs a set of possible states and transitions between these states, together with a small set of "... global contextual variables that help determine the next state". Processing takes the smallest unit of utterance that can be interpreted within a given context by applying a set of basic interpretative rules.

In attempts to achieve robust spoken language understanding, syntax has also been used to deal with recognition failures and ungrammatical inputs. For instance, Smith (1998) describes an approach whereby an ungrammatical output from the speech recognizer is transformed into possible grammatical interpretations through trying various word insertions, deletions and substitutions. Each of these transformations will have an associated cost (for instance, the cost of substituting "six" for "fix" would be low as they sound so similar) and the lowest cost solution is preferred. While this approach may relieve the constraints at the lower (lexical and recognition) levels, the language model (or grammar) used will still produce constraints at the syntactic level.†

In general, habitability could be increased by allowing flexibility in the syntactic structures that can be accepted. However, enforcing strict syntactic constraints in speech systems can aid the recognition process by reducing the perplexity (or branching factor) of a language (Schmandt, 1994). There is thus a potential trade-off between syntactic habitability and recognition performance. This effect has been demonstrated in a study on syntactic constraint in speech systems by Murray, Jones and Frankish (1996). Their system vocabulary in this study consisted of three distinct classes of word. In one condition they enforced a strict syntax where the three classes of word could only be spoken in a particular order. While this increased recognition performance compared to a no-syntax condition, overall performance was reduced because users frequently failed to operate within the required syntactic constraints. Murray *et al.* (1996) optimized performance by implementing what they called a partial syntax, where the syntactic rules allowed more flexibility in word order. Recognition performance here was improved compared to no syntax, while user inputs were more likely to be appropriate than in the full syntax condition.

2.4. LEXICAL CONSTRAINT

Lexical constraint refers to restrictions in terms of the individual words which are acceptable. For example, in the home banking domain it may be acceptable to refer to the balance of an account using terms such as "balance", "total" and "sum" but not using a less well-known synonym such as "reckoning".

As discussed above, one approach to increasing lexical habitability is to allow many synonyms for each interface object or action within the system vocabulary (Furnas *et al.*, 1987). For example, a system may be programmed to accept the synonyms "Oh", "zero" and "nought", for the digit 0. Unfortunately, such a strategy would be expected to

†Note that other approaches to robust parsing produce an analysis of the meaning of an utterance without the strict requirement that a grammatical interpretation of the utterance has been found first (see e.g. Allen, Miller, Ringger & Sikorski, 1996).

significantly increase the number of recognition errors if implemented in a speech recognition system, as it would increase the scope for confusions between vocabulary items. For example, the words "Oh" and "nought" are easily confused with other digit words and better performance is achieved by only accepting "zero". In an experiment using speech input, Robbe, Carbonell and Dauchy (1997) implemented a system lexicon where synonyms were excluded. They found that all users resorted to words outside this vocabulary at least once during the trials. One approach to deal with this kind of behaviour would be to limit the number of aliases to those words which people use most frequently during trials of the system. However, this approach is still unlikely to produce the kind of discriminable vocabulary set which will maximize recognizer performance. Vocabulary design can therefore be seen in terms of achieving a trade-off between user behaviour and system performance.

2.5. RECOGNITION CONSTRAINT

The final level of constraint that we define is recognition constraint. This refers to limitations at the level of recognizing spoken words. For instance, the same sequence of spoken words may or may not be recognized depending on various factors such as the speker's accent, the speech rate, the type of microphone used, whether the speaker pauses between words, the background noise level, etc. (Ainsworth, 1988; Schmandt, 1994).

Recognition errors are a major source of constraint in interactions with speech systems. There are several broad categories of recognition failure: insertions, rejections and substitutions (Williamson & Curry, 1984). Insertions occur when spurious noise is falsely recognized as a valid vocabulary item. Rejections and substitutions occur when a valid vocabulary word has been spoken but is not recognized correctly. In the case of rejections, no recognition is returned. With substitutions a different vocabulary item is returned. The occurrence of these errors within speech-ased systems introduces additional problems for the design and use of this technology.

The causes of recognition failure vary and system designers have tried a number of approaches to deal with them. The resulting system design alternatives produce varying degrees of constraint over how users can speak. A good example is the difference between isolated, connected and continuous word recognition systems. Isolated and connected word recognizers were developed to eliminate the complex computational problems of detecting word boundaries in speech and recognizing words despite changes in their sound due to co-articulation. However, while these methods can improve recognizer performance, they introduce significant constraints over how users can speak. With an isolated word system users must pause noticeably between words; with a connected word system they must be careful not to co-articulate adjacent words. Similarly, speaker-dependent systems have been developed to overcome the problem of words sounding different when spoken by different people. However, these systems introduce the constraint that they can only be used by those who have previously trained the vocabulary. As at the lexical level, designers introduce constraints with the aim of improving system performance, but this can lead to problems if users do not obey these constraints.

2.6. SUMMARY

As Ogden (1988) points out, for a system to be habitable it must be so at all levels. The key problem is that while achieving habitability in manual input systems is possible by reducing constraint (particularly at the lexical and syntactic levels), in speech systems this approach can have potentially devastating effects on recognition rates. Since speech system habitability cannot be achieved through simply expanding the language available, other ways must be found of increasing the match with user language use. Baber (1993) suggests that users' spoken exchanges with speech recognizers are influenced, to some extent, by the desire to develop an appropriate model of their dialogue partner. Given the potential ambiguity in defining an ASR device as a dialogue partner, people will use a variety of cues to do this. One important cue may simply be the knowledge that they are speaking to a computer. Several studies have demonstrated that users altered their speech to a human operator when told they were speaking to a computer. For example, Luzzati and Néel (1987) found that users constrained the syntax of their utterances when they believed they were speaking to a computer, choosing more regular, complete and concise utterances than when speaking to a human (see also Richards & Underwood, 1984; Amalberti, Carbonell & Falzon, 1993). The design of the machine prompts will also be expected to have a significant affect upon the selection and use of words. A simple exploitation of this effect is the use of bi-directional grammars which ensure that the system does not output any words that it cannot also recognize as input. Research has also investigated the design of explicit prompts to elicit the required vocabulary set (Kamm, 1994; Yankelovich, Levow & Marx, 1995). However, it seems that apparently unambiguous prompts need not eliminate out-task vocabulary. For instance, Kamm (1994) reports data from one speech application where only 54.5% of users' responses to the question, "will you accept this call?" were in the form of an isolated "yes" or "no". The alternative prompt, "Say "yes" if you accept that call, otherwise say "no"", resulted in 80.5% of user responses being in the form of an isolated "yes" or "no". Baber, Johnson and Cleaver (1997) investigated people's choice of words when speaking to a speech-based automated-teller machine. They found that the set vocabulary played some role but did not completely determine the choice of words. The study used various forms of visual and auditory feedback to prompt the users, and also involved a period of familiarization with the set of command words that the speech recognizer would use. Thus, one might anticipate that the users would be sufficiently familiar with the small set of command words to minimize the possibility of use of "out-task" words. However, the results indicated that users will attempt to use "out-task" words, or will use legal words inappropriately, even when they "know" the vocabulary. There was a relationship between type of task and the variation found in choice of words; some tasks produced little variation across conditions and others produced significant variation. Screen design also seemed to have a bearing on choice of words, but did not completely determine it, i.e. people did not simply copy what was on the screen. On the other hand, the addition of auditory feedback led to far less variation in choice of words, which suggests that auditory feedback reinforced the selection of certain words.

The Baber *et al.* (1997) work suggests a number of parameters which may influence user language choice, and hence habitability. However, more research is needed on the issue of habitability. On a theoretical level it is important to establish whether the

distinction between levels of habitability adds anything useful to our understanding of interactions with speech systems. For instance, is there evidence for Ogden's (1988) proposition that when a user's input fails they will not know which level of constraint has been violated? More importantly, what are the implications of this for the user's subsequent behaviour with the system, and hence their chance of eventual success? Another important theoretical issue is the extent to which habitability contributes to users' overall satisfaction with speech systems. It can be hypothesized that failure on an input, combined with a lack of understanding of the reasons behind that failure, would be more frustrating than input failure alone (Yankelovich *et al*., 1995). In addition, assuming that habitability is shown to be an important factor in speech system usability, research is needed to establish how it can be improved.

In this paper, we present two studies which start to explore the issue of habitability in speech system dialogues. In both studies users interacted with speech input systems which had been designed to impose particular sets of constraints on what inputs could be accepted. Of interest in the studies was how well users were able to operate within these constraints (i.e. the degree of dialogue habitability) and how behaviour was affected by features of the interface such as the output modality and prompt type. In the first study visual output was used to show the words and phrases which could be input to the system at any point within the interaction. Making the constraints visible in this way might be expected to lead to user inputs which the machine can accept (i.e. habitability). The reported study investigated the extent to which this is actually the case. The second study was an experimental comparison of two dialogue styles which vary in the extent to which they make the system constraints explicit. Comparisons between user behaviour, user subjective opinion and overall performance are used to explore the relationship between habitability and dialogue style. The key aims of the research are to further our theoretical understanding of the role of habitability in speech system interactions and to provide some initial guidelines for developers on how to increase speech system habitability.

In the studies reported in this paper, the required constraints were imposed through the use of fairly simple speech recognition technology. The desired system behaviour was simply that user inputs should fail if they violated the system constraints (at any level), or if they were misrecognized. This behaviour was achieved through the use of pre-designed sets of the words or phrases which could be accepted, which varied according to the user's position within the dialogue. While this approach is far from representing the cutting edge of speech system design, it does have the advantage of allowing a high level of control over the constraints which are imposed, making interpretation of the experimental results more straightforward. Habitability, by definition, is about achieving a balance between system and user behaviour. By maintaining strict control over system behaviour at this stage in our research, we hope to isolate some of the determinants of user behaviour. This reflects our emphasis on human factors issues rather than on technology design for its own sake. We recognize that more complex (or "intelligent") system behaviour is possible and in the discussion section of this paper we shall go on to consider how additional types of system behaviour might be investigated in future research.

TABLE 2
Constraints in study one

Constraint type	ATM system constraints	User cues
Semantic	Services available limited to account balance; cash with/without receipt; order chequebook or statement	User informed of all available sevices through system output (see dialogue level). Conceptual and functional constraints are implied through tasks given to users in the study (see Section 3.2)
Dialogue	Dynamic changes in available services occur as a result of current position within interaction	Display indicates current temporal position within the interaction and the currently active command words/phrases
Syntactic	Constrained at lexical level	See lexical level
Lexical	Total of 14 vocabulary items (eight single words, six short phrases)	Users are exposed to full vocabulary during recognizer training. Vocabulary items visible on-screen
Recognition	Speaker dependent. Isolated-word recognition	Users train the system prior to use, three similar pronunciations of each vocabulary item needed before training is complete

3. Study one: habitability in a speech-input/visual-output system

3.1. INTRODUCTION

A study was carried out to investigate the behaviour of users during the trial of a prototype speech-driven automated teller machine (ATM). The ATM interface used was based on current manual input systems, but instead of pressing keys situated adjacent to the ATM screen, users were required to read out the diplayed menu commands. This application can be analysed in terms of the levels of constraint described above. Table 2 shows what the actual system constraints are at each level and also what cues are available to inform users of these constraints.

Looking at Table 2 it appears that constraints are operating throughout all of the levels identified. However, in all cases the cues available from the system (and in some cases from the design of the study) unambiguously inform the user about what constitutes acceptable input. It was therefore hypothesized that this speech application would be habitable in the sense that users will be able to interact with it, without straying outside the language boundaries imposed. The study reported here investigated this proposal through observation of user interactions with the ATM. The spontaneous errors in users' inputs to the system were of particular interest, as any such behaviour indicates a departure from habitability. The human error results were analysed qualitatively in order to add to our understanding of what constitutes habitability in this type of application.

3.2. METHOD

3.2.1. Participants. Twenty-two people participated in the trials. Half were recruited from a local further education college, and half from amongst the staff and postgraduate population at Birmingham University. All participants were within the age range 16–35. None had any previous experience with speech recognition systems. All had experience of using manually operated ATMs.

3.2.2. Application design. The application represents a command type of interaction, with the available commands displayed as menu options throughout the interaction. The first screen simulated speaker verification for entry into the system, by requesting the user to speak a word displayed on the screen. The word was randomly selected from a set which had been previously trained, the idea being to introduce the participants to the notion of speaker verification (i.e. use of the ATM requiring card and speech, rather than card and PIN on conventional ATM). The main menu screen was then displayed, this offered five menu options: "withdraw cash", "cash with receipt", "balance", "order statement" and "order chequebook". If the user said either "withdraw cash" or "cash with receipt" then a further screen would be displayed on which the numbers "10", "20", "50", "100" and "200" appeared, along with the further option of "your choice" (which was not active). After a user spoke one of these values a screen was displayed which stated that the cash would be dispensed shortly. If the user had previously said "cash with receipt" the screen also informed them that a receipt would follow. The menu options available from this screen were "service" and "quit". Saying "service" would return the user to the main menu, saying "quit" would end the interaction. Similarly saying "balance" while in the main menu caused a further screen to be displayed showing a cash balance and the menu options "service" and "quit". If either the "order chequebook" or "order statement" options were chosen, the resulting screen informed the user that the chequebook or statement would be dispatched, and offered the menu options of "service" and "quit". In all cases only the menu options displayed were active. No confirmation or error correction facility was included.

It is worth noting that the options presented to users reflect "standard" ATM design in the UK and the initial screen designs were intended to be as familiar to users as possible. The tasks that users were asked to perform were designed to both reflect "standard" ATM use and require the users to read through the options and to select one appropriate to the task.

3.2.3. Procedure. Participants were first given a demonstration of the use of the speech-controlled ATM by the experimenter (cf. Baber, Stammers & Usher, 1990). They then trained the speech recognizer. This entailed speaking each vocabulary item 3 times into the recognizer when prompted to do so. If the three repetitions were not sufficiently similar to each other, the user was prompted to repeat the item a further 3 times. Participants then spoke each vocabulary item to check the recognition. Any vocabulary items showing persistent machine errors at this stage were retrained. The use of a speaker-dependent recognizer requires some discussion. It was felt that having participants "train" the recognizer would enable the experimenters to minimize the potential problems of recognition constraint, while also providing a means of familiarizing participants with the vocabulary set that would be employed.

Participants were observed interacting with the ATM in groups of 5–6. This was done in order to simulate real cash machine use by having a queue present behind the user. Each participant was given a card stating the services which they should obtain from the ATM. Each participant was given two goals to achieve, for example "check your balance and withdraw £10 (with receipt)". Participants were informed before starting that if their inputs were not recognized at first to try repeating the command. During the experiment if a command was not recognized within six repetitions the command was chosen manually by the experimenter and the dialogue continued. After performing their trial participants went to the back of the queue behind the next user. The question of whether the participants should undertake the trial individually or as a group was considered during the design of the study. It was felt that having participants stand in a queue would provide an element of ecological validity that individual trials would have missed. The queue functioned as a means of providing social context for task performance and also as a means of inducing (very mild) stress on the participants. The drawback of this approach is that the participants may learn from the behaviour of their peers. However, the distribution of "error" responses across the separate trials suggests that this was not a significant factor in the current study.

3.2.4. Apparatus. The recognizer used was the Voice Navigator for the Apple Macintosh, an isolated word, speaker-dependent system. The acceptance criteria used by the recognizer were such that rejections were the most common recognizer error, and substitutions and insertions were highly unlikely to occur. The recognizer output was used to control a Hypercard program running on a Power Macintosh. The screen of the Macintosh was placed within a mock-up of an ATM machine, complete with manual input pad and slots for insertion of cash card and withdrawal of cash. This screen displayed the menu options as described above. Participants spoke into a microphone mounted on the ATM.

3.3. RESULTS AND CONCLUSIONS

Fifteen out of the 22 participants made at least one error in their inputs to the system (i.e. said words or phrases which were not available) and overall 20 user errors were observed. These results indicate that, contrary to expectations, this application was not entirely habitable for first-time users.

The results from this study will now be dealt with qualitatively in order to give an overview of the types of user errors which occurred. User errors can be categorized according to the levels of constraint defined in Table 1.

3.3.1. Violations of dialogue constraint. Dialogue violations occur when participants used valid vocabulary items, but at points in the dialogue where these were not active. Five of the 22 participants made this type of error in interacting with the system. Interestingly, all of these errors appeared to show users trying to jump ahead in the dialogue to achieve their next goal. Four errors were where users asked for cash when on a secondary screen (e.g. displaying balance) and the available options were "service" or "quit". One participant said "order statement" on the speaker verification screen.

Further analysis of the instances when this type of error behaviour occurred is instructive. In the application used there are two main types of vocabulary item: words which command the system to carry out an action (e.g. display a balance or dispense cash) and words which allow the user to navigate between screens. The navigation between screens is a feature of "standard" ATM interactions and in this application provides a means of controlling the number of options available at any one time, which in turn can reduce the potential for recognition error. When these two types of vocabulary item are examined in relation to users' task goals, there is a clear distinction between them. The system commands map directly on to users' goals while the navigation commands do not. Tellingly, the dialogue constraint violations observed only occurred when the available vocabulary served a navigation function. This suggests that users will follow dialogue structures to the extent that these match their own goals, but where there is a divergence between user and system goals, the potential for error increases. This effect may be due to the relative salience of system operators (words) which fulfil user's high-level goals compared to operators which fulfil system navigation goals. Alternatively, the behaviour may represent user attempts to increase interaction speed with the system by trying to avoid system navigation.

3.3.2. Violations of syntactic and lexical constraint. Throughout the study users were required to speak only those words or phrases which were displayed on the screen. However, there were three instances where participants added the word "pounds" to a number when requesting cash, and two instances where participants omitted the word "order" from the phrase "order statement". This suggests that although these participants appreciated the dialogue level of constraint (by restricting their choice to one of the available meanings) they did not appreciate the syntactic constraint which required them to say exactly what appeared on the screen.

3.3.3. Violations of recognition constraint. Observation of the users also indicated several instances of recognition constraint violations. These typically involved users pausing noticeably within phrases (for instance, "order [pause] statement"). Although the number of these violations was not counted due to the difficulty of measuring whether a distinct pause had occurred, they would have had a negative impact on recognition accuracy, and indeed rejection errors were frequent. This user behaviour indicates a failure to appreciate the recognition limitations of the device, despite the fact that during system training the user had been required to produce three similar versions of each vocabulary item, each with no pause in the phrase.

3.4. CONCLUSION

This study demonstrates the importance of considering habitability in speech input systems. Unlike manually operated ATM systems (which have a finite number of controls), speech systems do not provide physical constraint over what the user can input. This lack of physical constraint increases the potential for user errors, where inputs do not match what the system can accept.

The study also demonstrates that a good match between user and system is not achieved simply by showing system options. Users' goals are also important. When user errors were examined in relation to the goal structure of the task it was found they could

be explained as users trying to pursue their own higher goals rather than the actions required to navigate the interface at that point in the interaction. This result suggests that matching dialogue options to higher level user goals would improve habitability (in terms of reducing the likelihood of user error). For instance, instead of navigation commands on a screen displaying a users' balance, they could be given options such as withdrawing cash or ordering a printed statement. This strategy would also have the potential advantage of reducing transaction time.

Analysis of the violations of syntactic/lexical constraint indicate that some users' did not perceive these constraints as being as strict as they actually were. Their utterances included valid lexical items, but not in the format accepted by the recognizer. This effect was observed despite the fact that users trained the system prior to use with three repetitions of the exact phrase needed. This result suggests that even when system options are displayed on screen, some variation in the syntactic form with which they can be entered will be beneficial in terms of increasing system habitability.

The speech system used in this study was highly restrictive in terms of what it could accept. It was speaker dependent, with a limited vocabulary and had no dialogue management facilities. Nevertheless, the findings have relevance to the design of more contemporary systems as they reveal user behaviour with the particular type of cues to system constraint given in this application. In the study there was a direct match between what users could see on the screen, the recognizer training they had engaged in, and what the system could actually accept. The deviations from acceptable input observed therefore indicate areas where changes in system design could increase habitability. The analysis of user behaviour in terms of the different levels of constraint should help to focus this design effort appropriately.

4. Study two: habitability in a speech-input/speech-output system

4.1. INTRODUCTION

The second study considered user interactions with a speech-input/speech-output system. As in study one the domain was banking, but in this case a telephone home banking application was developed. Previous work by Baber *et al.* (1997) found that a combination of auditory and visual feedback in a speech-based ATM led to less variation in choice of words than a similar ATM with only visual feedback. However, it is unclear whether it was the presence of auditory feedback, or the combination of the two types of feedback, which led to this apparent improvement in habitability. The current study addresses this issue by investigating user behaviour in response to only auditory computer output.

Two alternative dialogue styles were included in the study in order to investigate the effect on habitability of varying the level of system constraint. One dialogue style used spoken menus and users were restricted to answering with one of the words or phrases presented to them in the menu. The other dialogue style used spoken queries where the prompts gave no explicit cue as to the level of syntactic or lexical constraint operating within the system. The two dialogue styles represent extremes for the purposes of comparison; many contemporary systems might well use a mixture of these approaches.

TABLE 3
Constraints in the menu-style application

Constraint type	ATM system constraints	User cues
Semantic	Services available limited to account balance; bill payment; order chequebook or statement	User informed of all available sevices through system output (see dialogue level). Conceptual and functional constraints also implied through tasks given to users in the study (see Section 3.2)
Dialogue	Dynamic changes in available services occur as a result of current position within interaction	Spoken prompt indicates current temporal position within the interaction and the currently active command words/phrases
Syntactic	Constrained at lexical level	See lexical level
Lexical	10 words or short phrases	Currently available vocabulary items presented in spoken prompt
Recognition	Connected-word recognition, system only recognizes pre-determined single words or short phrases	Spoken prompt provides example of acceptable pace of speaking, pause length, etc.

The system constraints which were operating in each dialogue style, and the cues available to users, are summarized in Tables 3 and 4.

Observations were made of user behaviour with the different dialogue styles, in particular looking for any evidence of lack of habitability at the various levels of constraint identified. In addition, overall performance data and subjective responses were collected and compared. It was hypothesized that the menu-style dialogue (where constraints are explicit) would be more habitable than the equivalent query-style dialogue. It was also hypothesized that users would prefer the more habitable style. Also of interest was how behaviour with the spoken menu prompts used in this study compared with behaviour with the visible menu prompts used in the first study.

4.2. METHOD

4.2.1. Participants. Thirty participants were recruited from the Nottingham area (mean age 23 years, 10 months). There were 16 males and 14 females. All had used ATMs but none were experienced users of telephone banking systems. The participants were all UK nationals and were paid UK£5 for their participation.

4.2.2. Application design. Both dialogues began with a prompt welcoming the user to the "Bank of Nottingham" home banking service. They were asked to input a user number and PIN at this stage (though note that any input was accepted for this step). The next prompt for the menu-style dialogue was: "every time you are asked to say

TABLE 4
Constraints in the query-style application

Constraint type	ATM system constraints	User cues
Semantic	Services available limited to account balance; bill payment; order chequebook or statement	Semantic constraints implied through tasks given to users in the study (see Section 3.2)
Dialogue	Dynamic changes in available services occur as a result of current position within interaction	Spoken prompt indicates current temporal position within the interaction and provides a cue to the current level of conceptual constraint
Syntactic	Range of syntactic forms allowed for each service; actually constrained at lexical level	Initial prompt instructions inform user that system can accept single words and short phrases
Lexical	27 words or short phrases	None
Recognition	Connected word recognition	None

something you will be given a list of words to choose from. Say the option you want after the list has finished". Query-style dialogue users were told: "the system can recognize single words and short phrases. After each question say the service you want. For example after the question "which service do you want?", say "order chequebook".

In the menu-style dialogue users were next presented with the prompt: "Say one of the following services; balance, statement, others". Following recognition of either "balance" or "statement", users were given the prompt: "choose an option for the [balance/statement] you require; current, savings or both". Following recognition of "others" users were given the next level menu of "Choose one of the following services; chequebook, pay bill or other". If the recognizer was unable to process an utterance (rejection error) the user was given the message "the system did not recognize an input" and the previous prompt was replayed. After three consecutive fails users were given the message "the system cannot complete this service, do you want to try again; yes or no".

In the query-style dialogue users were presented with the prompt: "which service do you require?". The same services were available as in the menu dialogue (i.e. balance, statement, new chequebook and pay bill) and for each of these a range of different ways of phrasing the request was included in the system lexicon (defined via a pre-study simulation in which a small sample of users could answer freely to this question). For example, if users wanted a balance they could say "hear balance", "balance", "hear my balance", "current account balance" or "savings account balance". If a service was recognized without an bank account being specified then users were given a second prompt asking, for example, "which account would you like a balance for?". Fails were dealt with as in the menu condition. After three consecutive fails users were given the message: "the system cannot complete this service, choose another service or say quit".

4.2.3. Procedure. Participants were given an instruction sheet detailing four tasks to be performed: finding out the savings account balance, paying off a credit card bill by transferring money from savings, and obtaining statements for both the savings and the current accounts. They were instructed to speak clearly and naturally, but were given no further instructions on how to speak or what to say. Participants were free to quit the interaction at any time by saying "quit" (equivalent to putting the telephone down). After their interaction participants were asked to complete a questionnaire.

4.2.4. Apparatus. The dialogues were programmed in Visual Basic 5.0 running on a Pentium II PC. Speech recognition was achieved using Dragon Dictate 3.0 (British Edition) and was integrated into the Visual Basic project using Dragon X-Tools. Speech output was provided through recorded human speech stored as WAV files. The same female voice was used for all system prompts. A head-mounted VXI Corp Parrot 10.3 microphone/headphone was used throughout.

4.2.5. Experimental design. The experiment used a between subjects design with 15 participants using each dialogue style. The experimental groups were balanced for gender and age. Data were recorded on overall success (proportion of tasks completed) and on user error behaviour with the systems. Subjective responses were collected using a questionnaire designed to assess user opinions on various aspects of speech-based systems. This questionnaire contains a number of items designed to assess habitability and also includes questions on the user's general opinion of the interaction and likelihood of using it if available (Hone & Graham, 2000).

4.3. RESULTS

4.3.1. Task success. The mean number of sub-tasks completed in the menu condition was 2.6 (S.D. 1.40), and in the query condition it was 1.1 (S.D. 1.62). An independent sample t-test revealed a significant difference between these two means [$t(28) = 2.766$, $p < 0.05$], indicating that participants were significantly more successful with the menu-style dialogue than with the query-style dialogue.

In the menu condition, failures to complete tasks resulted almost entirely from recognition failures (rejection errors). In the query condition, failures to complete tasks were due to both recognition failures and users inputting out-task vocabulary items or phrases. Recognition performance did not differ significantly across the two conditions.

4.3.2. Subjective responses

4.3.2.1. Overall ratings. Three questions were designed to elicit responses about the general usability of the system: "Overall I think this is a good system," "The system was easy to use" and "I would use this system." The mean responses are shown in Table 5. Note that all responses were given on a 7-point scale where 1 = strongly agree and 7 = strongly disagree.

These results indicate that the menu style was rated as significantly better than the query style and as significantly easier to use.

TABLE 5
Subjective ratings

Question	Mean rating				
	Menu	Query	t	df	p
Overall I think this is a good system	3.4 (S.D. 1.6)	4.9 (S.D. 1.5)	−2.72	28	<0.05
The system was easy to use	3.1 (S.D. 1.8)	5.0 (S.D. 1.3)	−3.29	28	<0.01
I would use this sysem	4.4 (S.D. 2.1)	5.1 (S.D. 1.7)	−1.07	28	>0.05

TABLE 6
Habitability ratings

Question	Mean rating				
	Menu	Query	t	df	p
I sometimes wondered if I was using the right word	3.3 (2.1)	6.0 (1.4)	−4.20	28	<0.001
I always knew what to say to the system	2.1 (1.3)	5.7 (1.6)	−6.99	28	<0.001
It is clear how to speak to the system	2.1 (1.3)	4.7 (1.7)	−4.60	28	<0.001

4.3.2.2. Habitability ratings. Three questions were included which were intended to directly address habitability. Table 6 summarizes the ratings obtained and the results of independent sample t-tests comparing the means for each question.

The results shown in Table 6 indicate that users found the menu style significantly more habitable than the query style.

4.3.3. Violations of system constraints. There were a number of instances where the system rejected a user input, within both the menu and the query conditions. As in study 1 these can be examined in relation to the different levels of dialogue constraint.

In the menu condition there were no examples of users violating dialogue constraint. Spoken prompts and questions from the system were always responded to with appropriately constrained utterances. Unlike study 1 there were no examples of users trying to bypass the dialogue constraints in order to achieve task goals more quickly. Violations of syntactic/lexical constraint were negligible. In fact, only two such violations were noted across all the trials, with the users in both cases being heard to say "other" rather than

"others" as given in the prompt choices. On both of these occasions the users re-entered the word as it was given in the prompt. The majority of errors in the menu condition were thus due to violations of recognition constraint.

In the query condition there were examples of violations at several levels of constraint. Three were examples of users simply not knowing what to say (e.g. either saying nothing, or in one case saying "I don't know what to do"). There were no clear-cut examples of violation of lexical constraint, with all user utterances (except those just mentioned) containing at least one word from the system lexicon. The remaining instances of failure for the query condition are categorized below.

4.3.3.1. Violations of dialogue constraint. Analysis of user behaviour in the query condition revealed a number of violations of dialogue constraint (12 out of a total of 78 rejections of user inputs). In all cases these were instances where the prompt asked users to choose a service and users instead responded with a type of bank account (savings or current). Interestingly, all of these violations followed on from a system rejection of a user utterance which had included a valid service. This suggests that users may have interpreted these rejections as evidence that they had chosen the wrong conceptual category for their original utterance and therefore retried using a response from an alternative conceptual category. In attempting to recover from these failures there were nine instances where users correctly reverted to asking for a type of service, and three instances where users wrongly repeated or rephrased a bank account name.

4.3.3.2. Violations of syntactic constraint. In the query condition there were 23 instances of syntactic violations. In these instances users spoke relevant lexical items, but within sentences or phrases which were not acceptable. After 19 of these failures users had the chance to try again. Of these retries three were repetitions of the original utterance (suggesting users wrongly attributed the failure to recognition constraint), seven were rephrases of the original utterance (suggesting users correctly attributed the failure to syntactic constraint) and nine were changes of topic where users requested a different service. This last type of behaviour never occurred on a first failure, suggesting that they represent users giving up on the initial service and going on to the next one.

4.3.3.3. Violations of recognition constraint. In the query condition 40 of the 78 input failures observed (i.e. about half) could be categorized as recognition failures. Of these, 35 presented the user with an opportunity to try again on the next turn. Given that a recognition failure had occurred, the appropriate strategy at these points would have been for users to repeat the original utterance. Instead, it was found that only 17 of the re-entry attempts were repetitions of the original valid utterance while 18 were new formulations of the request. This result suggests that overall participants were no better than chance at reacting appropriately to recognition errors in the query condition. These data can be broken down further according to the type of prompt to which the user was responding. The results of this analysis are shown in Table 7.

The data shown in Table 7 suggest that there are important differences in user behaviour depending on the type of prompt which is being responded to (note that these data do not meet the assumptions for chi-squared analysis and therefore these differences cannot be tested statistically). The pattern of results suggests that users become more likely to behave appropriately as the size of the potential response set is reduced.

TABLE 7
User behaviour after a recognition failure (by prompt type)

User strategy	Prompt restricts user to		
	A bank service	A type of bank account	Yes/no
Repeat first utterance	6	5	6
Re-phrase utterance	15	3	0

4.4. CONCLUSION

The high level of adherence to the required dialogue and syntactic/lexical constraints shown by participants in the menu condition suggests that spoken prompts provide a clearer cue to these constraints than similar visual prompts (cf. Study 1). This finding implies that habitability is influenced by the modality in which feedback and prompts are presented, as well as by the content of these machine outputs.

The comparison between the menu condition and the query condition revealed several important differences. First, the two styles were rated significantly differently on questions designed to assess habitability. Users were considerably less sure of what words to say to the query system and also of how to speak to the system. While this result is not surprising given that the menu system stated the exact words to say and the query system did not, the large differences in ratings confirm the importance of prompt design for habitability. The two styles were also rated differently in terms of ease of use and general quality. These differences did not, however, translate into differences in the rated likelihood of using the system (probably because participants did not, on average, want to use either of the two systems). Overall the menu style system was more effective than the query system, with participants in the menu condition completing more sub-tasks than those in the query condition.

User behaviour with the query dialogue supports the hypothesis that where habitability is reduced users will have difficulty in knowing which level of constraint has been violated when confronted with a failure. For example, the most frequent cause of failure was recognition failure, and yet users only used "appropriate" error recovery strategies in half of these cases. In the remaining cases behaviour indicated that users had misattributed their failure to a violation of syntactic constraint. Overall, rephrasing an utterance (i.e. changing the syntax) accounted for over half of users' re-input attempts, while syntactic errors accounted for less than a third of actual failures.

5. Discussion

The research reported in this paper supports the idea that habitability is a key issue in the design of speech-based systems. Changes in the prompt wording in Study 2 produced clear differences in the subjective habitability ratings given to the dialogue (for example, whether the user knew what to say to the system). These differences were accompanied by better performance in the more habitable condition in terms of the number of sub-tasks

completed, and by a user preference for the more habitable version. These results indicate that habitability may be as important a determinant of speech system usability as recognizer performance. If users do not know what to say to a system they will not succeed in their tasks.

The data reported in this paper also provide further insight into the factors that can determine habitability. At the start of this paper a habitable computer language was defined as one in which users can express themselves without straying over the language's boundaries into illegal ("out-task") language. It was suggested that using a visual display to show all the currently valid speech inputs would produce a highly habitable system, as such a display should unambiguously bridge the gap between system capabilities and user expectations. However, Study 1 showed that the use of such an interface is not sufficient to ensure a habitable system. Participants in this study sometimes deviated from the syntax displayed on screen. This behaviour may be due to users either ignoring or misreading the display, an explanation which strongly relates to previous work showing that people do not accurately monitor visual feedback from a speech recognizer (Baber, Usher, Stammers & Taylor, 1992; Noyes & Frankish, 1994). It seems that the users' model of the system's capabilities was that their utterances should overlap with, but not necessarily match, the displayed input structures. Some participants in this study also violated the dialogue level constraint. These participants tried to jump ahead within the interaction by asking directly for the next service, rather than following the displayed commands which allowed navigation to the appropriate screen within the system. Analysis of this behaviour highlighted the importance of user goals in determining behaviour. The practical implication of this result is that habitability could be increased by creating a closer link between dialogue structures and user goal structures. Designers should thus avoid the use of speech commands simply to navigate within an interface structure, and instead concentrate on making the next services (user goals) available as options. This highlights the need for careful analysis of user needs prior to implementing a speech system.

Explicit prompts, giving the currently available input vocabulary, were also used in the menu condition in Study 2, but in this case they were presented through spoken rather than visible system output. Here users were much better able to keep within the constraints of the system language. This result suggests that computer output modality may be a determinant of speech system habitability, with the speech output system in this case being more habitable than a comparable visual output system. It has often been suggested that speech input systems should use spoken output in order to preserve "stimulus response compatibility" (e.g. Wickens, Sandry & Vidulich, 1983), but strong empirical evidence in support of this contention has not been forthcoming (Baber, 1991). The research reported here suggests one tangible benefit of spoken output is that it can serve as a stronger cue for dialogue, syntactic and lexical constraint than equivalent visual output. However, there are problems with the approach of providing the available options in the form of a spoken menu. First, it is time consuming. Second, only a limited number of options can be presented pre prompt [Schumacher, Hardzinski & Schwartz (1995) suggest a maximum of four items] in order to fit in with the capacity constraints of human working memory. This means than several levels of menus will be needed when the number of services available exceeds four. Thus, increased habitability comes at a significant time cost. Research by Brems, Rabin and Waggett (1995) suggests that it

may be possible to offset this cost to some extent through the use of "barge-in" where the user is allowed to interrupt the recognizer output. Presenting the most frequently required options first will also improve interaction speeds.

An alternative is to provide prompts which aim to constrain the meaning of the user's utterance at that point in the interaction (dialogue constraint level), but which do not specify the required input at the syntactic or lexical level. This approach was also tested in Study 2 with the query-style dialogue. Although a number of instances were observed where users disobeyed the dialogue constraint, these never occurred on a first attempt at accessing the system services. This finding suggests that the query prompts provided a fairly strong cue to dialogue constraint, but that uncertainty remained for some users, leading them to try an alternative meaning with their next input. Users seem in this case to be engaging in problem-solving behaviour, testing the hypothesis that their inputs failed because they had tried the wrong meaning.

With query-style prompts there were several instances where users spontaneously spoke utterances which disobeyed the syntactic constraint of the system. When these inputs failed there was a good chance that users would behave appropriately and try an alternative syntactic structure. However, this error recovery strategy was over-generalized and was also applied in situations where the original utterance had followed the appropriate syntax. Such a strategy will on average decrease performance, as not all the rephrased utterances will be acceptable. Overall the most common reason for an input failing was a recognizer rejection, indicating a failure at the recognition level, but in half of these cases users tried to recover in an inappropriate way by trying a new syntactic structure, rather than by repeating the original utterance. These results support Ogden's (1988) proposal that users will have problems identifying which level of system constraint has been violated when an input is rejected. The results also present the picture of a user as an active explorer of system capabilities. When one utterance fails they are likely to try a different one and keep varying their input until they get a response. This observed behaviour fits in with Baber's (1993) suggestion that users' spoken exchanges are influenced by the desire to develop an appropriate model of their dialogue partner. In the current study users appeared to be gathering data in order to build up their system model. The resulting models overestimated the role of syntactic constraints in preventing acceptance of their inputs, and underplayed the role of recognition constraints. A key question for designers is how to allow users to develop a more appropriate system model, without actually specifying the full range of acceptable utterances.

We would argue that in order to develop a more appropriate system model, users need to know why their utterances have failed. Specifically, there should be more indication of which level of failure is involved. On a very simple level it may be worthwhile warning users that speech recognizers are error prone devices, and may sometimes fail to recognize inputs which are acceptably worded. Potentially more useful would be dynamic changes to system feedback indicating the type of failure which has occurred. Although it will not always be possible for a speech recognizer to detect exactly why an input has failed, there may be some very useful cues in the input signal which could be analyzed and used to generate appropriate feedback. For example, the recognizer could detect loudness and accordingly tell the user to speak more loudly or quietly. In user trials of their spoken natural language dialogue system Smith and Hipp (1994) found it necessary for an experimenter to remain in the room with the users and tell them when

their utterances had failed because of a misrecognition error. The experimenters also made other comments relevant to habitability such as "the word X is not in the vocabulary" (lexical constraint) and "please keep your speech tone/volume/rhythm similar to the way you trained" (recognition constraint). The success of these comments in facilitating user interactions with the system demonstrates the utility of this approach, but ideally there should be no need for a human intermediary between user and system (Smith, 1997).

As well as informing interface design, analysis of user recovery behaviour according to constraint level in the way that has been done in this paper is potentially helpful in the design of algorithms for dealing with spoken language input. For example, algorithms could be designed to exploit the knowledge that users of spoken dialogue systems are likely to obey dialogue constraint on a first attempt but may not continue to do so following an input failure. It will be important to evaluate the impact which more "intelligent" processing of this kind has upon the models users develop of the system and hence habitability. The studies reported here used a straight-forward system model where all acceptable input words or phrases were pre-defined, and user inputs were simply rejected if they did not match these exactly, or if they were misrecognized. However, much contemporary research in the field of spoken natural language understanding is concerned with the creation of robust systems which are not limited in this way. Some of these approaches have already been briefly introduced above (see Section 2). The strategies aim to produce meaningful or grammatical interpretations of user utterances despite various types of failure (machine misrecognitions, ungrammatical utterances, use of out-of-task vocabulary, etc.). While processing of this kind reduces the probability of the system incorrectly rejecting a valid input, it also increases the probability of misinterpretations and of wrongly accepting invalid inputs. For this reason it is important to consider how the system acts on its interpretations. One approach, the "strong commitment" model (Allen *et al*., 1996), is for the system to always act as if its interpretation is correct. This relies on the system being able to recognize from subsequent user turns that the original interpretation was in fact wrong. At the opposite extreme, all interpretations can be explicitly checked with the user through the use of a verification sub-dialogue. Clearly, this approach has the drawback of being unnecessarily time consuming where the original interpretation is correct. Between these two extremes there is the option to selectively engage in verification sub-dialogues. Smith (1998) describes how parse cost and context information can be used in deciding when to verify, with the aim of minimizing both "under-verification" (incorrect meaning generated but no verification) and "over-verification" (correct meaning generated but still verified). Research such as this is rightly concerned with maximizing system effectiveness and efficiency, but there does not as yet appear to be much emphasis on what effect the resultant system behaviour might have on users. We would predict that both the strong commitment model and selective verification have the potential to reduce dialogue habitability because users may be confused when the system generates, and acts upon, incorrect interpretations of their utterances. This hypothesis would make an ideal topic for future research into the habitability of spoken language systems. One factor to investigate is the level, or levels, at which the robust parsing techniques operate; for example one can distinguish between techniques which aim to generate grammatical interpretations from misrecognized input and those which aim to generate meaningful

TABLE 8
Design guidelines to increase speech system habitability

Constraint type	Design guidelines
Dialogue	Consider auditory prompts rather than visual prompts Match dialogue structure to user goal structure
Syntactic	Allow small variations from displayed syntactic form in speech input/visual output systems Use keyword spotting Provide users with relevant feedback if multiword inputs are used when single words or short phrases are required
Recognition	Warn users that recognition failure is likely Provide relevant feedback if analysis of user inputs suggests that these have violated recognition constraints (e.g. too loud, too quiet, too fast etc.)

interpretations from syntactically ill-formed input. Another factor to investigate is the effect upon habitability of using the alternative methods of verification available (i.e. none, selective or all). The ideal design solution will be effective and efficient, but not at the expense of user satisfaction and habitability.

6. Conclusions

The key conclusions from the studies reported here are as follows.

- Habitable speech systems are preferred by users and lead to better performance.
- Spoken prompts increase habitability by providing a stronger cue to dialogue and lexical constraints than equivalent visual system output.
- A good fit between dialogue structure and user goal structure should increase habitability (particularly for visual output systems).
- Users will have problems identifying the level of constraint which has been violated when a failure occurs.

From these results it is possible to provide some guidance to the designers of speech interactive systems. This is summarised in Table 8. The guidance is organized according to the different levels of habitability defined in this paper. We hope that this framework will help to remind designers of the need to consider the constraints which their designs introduce at each of the different levels. Habitability will only be achieved when all the levels are considered in design.

The research reported in Study 1 was supported by a grant from NCR, while Study 2 was supported by the Engineering and Physical Sciences Research Council (Grant Reference: GR/L94710). The authors would like to thank David Golightly for his work on the project reported in Study 2, and Robert Graham and the anonymous referees for their useful comments on this paper.

References

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

ALLEN, J. (1995). *Natural Language Understanding*. Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc.

ALLEN, J. F., MILLER, B. W., RINGGER, E. K. & SIKORSKI, T. (1996). A robust system for natural spoken dialogue. *Proceedings of the 34th Meeting of the Association for Computational Linguistics*. Santa Cruz, CA, USA,

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.

BABER, C. (1991). *Speech Technology in Control Room Systems: a Human Factors Perspective*. New York: Ellis Horwood.

BABER, C. (1993). Developing interactive speech technology In C. BABER & J. M. NOYES, Eds. *Interactive Speech Technology*, pp. 1–17. London: Taylor & Francis.

BABER, C., JOHNSON, G. I. & CLEAVER, D. (1997). Factors affecting users' choice of words in speech-based interaction with public technology. *International Journal of Speech Technology*, **2,** 45–60.

BABER, C., STAMMERS, R. B. & USHER, D. M. (1990). Instructions and demonstration as media for new users of ASR. *Behaviour and Information Technology*, **9,** 371–379.

BABER, C., USHER, D. M., STAMMERS, R. B. & TAYLOR, R. G. (1992). Feedback requirements for ASR in the process control room. *International Journal of Man Machine Studies*, **37,** 703–719.

BREMS, D. J., RABIN, M. D. & WAGGETT, J. L. (1995). Using natural language conventions in the user interface design of automatic speech recognition systems. *Human Factors*, **37,** 265–282.

BUNT, H. C., LEOPOLD, F. F., MULLER, H. F. & VAN KATWIJK, A. F. V. (1978). In search of pragmatic principles in man-machine dialogues. *IPO Annual Progress Report*, **13,** 94–98.

FURNAS, G. W., LANDAUER, T. K., GOMEZ, L. M. & DUMAIS, S. T. (1987). The vocabulary problem in human-system communication. *Communications of the ACM*, **30,** 964–971.

HONE, K. S. & BABER, C. (1999). Modelling the effect of constraint on speech-based human computer interaction. *International Journal of Human Computer Studies*, **50,** 85–107.

HONE, K. A. & GRAHAM, R. (2000). Towards a tool for the subjective assessment of speech system interfaces (SASSI). *Natural Language Engineering*, **6,** 287–305.

KAMM, C. (1994). User interfaces for voice applications. In D. B. ROE & J. G. WILPON, Eds. *Voice Communication between Humans and Machines*, pp. 422–442. Washington, DC: National Academy of Sciences.

LUZZATI, D. & NÉEL, F. (1987). Linguistic behaviour brought about by the machine. *Proceedings of the European Conference on Speech Technology*, Vol. 2, pp. 401–404.

MURRAY, A. C., JONES, D. M. & FRANKISH, C. R. (1996). Dialogue design in speech-mediated data-entry: the role of syntactic constraints and feedback. *International Journal of Human–Computer Interaction*, **45,** 263–286.

NOYES, J. M. & FRANKSISH, C. R. (1994). Errors and error correction in automatic speech recognition systems. *Ergonomics*, **37,** 1943–1957.

OGDEN, W. C. (1988). Using natural language interfaces In M. HELANDER, Ed. *Handbook of Human–Computer Interaction*, pp. 281–299. Amsterdam: Elsevier.

RICHARDS, M. A. & UNDERWOOD, K. M. (1984). Talking to machines. How are people naturally inclined to speak? In E. D. MEGAW, Ed. *Contemporary Ergonomics*, pp. 62–67. London: Taylor & Francis.

ROBBE, S., CARBONELL, N. & DAUCHY, P. (1997). Constrained vs spontaneous speech and gestures for interacting with computers: a comparative empirical study In S. HOWARD, J. HAMMOND & G. LINDGAARD, Eds. *INTERACT '97*, pp. 445–452. London: Chapman & Hall.

SANDERS, M. S. & MCCORMICK, E. J. (1992). *Handbook of Human Factors*. New York: John Wiley.

SCHMANDT, C. (1994). *Voice Communication with Computers*. New York: Van Nostrand Reinhold.

SCHUMACHER, J. R., HARDZINSKI, M. L. & SCHWARTZ, A. L. (1995). Increasing the usability of interactive voice response systems: research and guidelines for phone-based interfaces. *Human Factors*, **37,** 251–264.

SHNEIDERMAN, B. (1998). *Designing the User Interface*: *Strategies for Effective Human-Computer-Interaction*. (3rd Edn.) Reading, MA: Addison-Wesley, Longman Inc.

SMITH, R. W. (1997). Performance measures for the next generation of spoken natural language dialog systems. *Proceedings of the ACL'97 Workshop on Interactive Spoken Dialog Systems*, pp. 37–40.

SMITH, R. W. (1998). An evaluation of strategies for selectively verifying utterance meanings in spoken natural language dialog. *International Journal of Human–Computer Studies*, **48,** 547–552.

SMITH, R. W. & HIPP, D. R. (1994). *Spoken Natural Language Dialog Systems*: *A Practical Approach*. Oxford: Oxford University Press.

WATERWORTH, J. A. (1982). Man–machine speech dialogue acts. *Applied Ergonomics*, **13,** 203–207.

WATT, W. C. (1968). Habitability. *American Documentation*, **19,** 338–351.

WICKENS, C. D., SANDRY, D. & VIDULICH, M. (1983). Compatibility and resource competition between modalities of input, output and central processing. *Human Factors*, **25,** 227–248.

WILLIAMSON, D. T. & CURRY, D. G. (1984). Speech recognition performance evaluation in simulated cockpit noise. *Speech Tech '84*, pp. 599–616.

WOLF, C. G., KASSLER, M., ZADROZNY, W. & OPYRCHAL, L. (1997). Talking to the conversation machine: an empirical study In S. HOWARD, J. HAMMOND & G. LINDGAARD, Eds. *INTER-ACT '97*, pp. 461–468. London: Chapman & Hall.

YANKELOVICH, N., LEVOW, G.-A. & MARX, M. (1995). Designing speech acts: issues in speech user interfaces. *CHI '95*. New York: ACM.

Paper accepted for publication by an Associate Editor, David Hill