‘The Sound of Silence’: A Preliminary Experiment Investigating Non-Verbal Auditory Representations in Telephone-Based Automated Spoken Dialogues[1]

David Williams

Vocalis Ltd., Great Shelford
Cambridge, UK[2]

Christine Cheepen

Dept. of Sociology, University of Surrey
Guildford, UK

Abstract

At the lexical level, a typical human-computer dialogue in an aural-only spoken language system consists of two stages, system output and user input. As with human-human conversation, a good proportion of turn taking clues are given by lapses in talk. Unfortunately, in telephone-based automated spoken dialogues, silences on the system’s part may not be so easily resolved. A pilot experiment examined the recogniser listening and processing states and showed that auditory icons representing these caused fewer incorrect user responses than the control condition. However, where system prompts explicitly requested a response, icons were not necessary if talkover was provided. Also, the effectiveness of auditory representations had a strong interaction with the expertise of the caller suggesting that expert users may require a period of acclimatisation to the use of sounds as they tend to listen to them due to novelty. Conversely, novice users with no experience acted correctly.

1 Introduction

At the lexical level, a typical human-computer dialogue in an aural-only spoken language system consists of two stages, system output and user input. As with human-human conversation, a good proportion of turn taking clues are given by lapses in talk. Unfortunately, in telephone-based automated spoken dialogues, silences on the system’s part may not be so easily resolved. These potentially ambiguous representations are resolved by a variety of means such as surrounding semantic, syntactic and prosodic information or physical gestures, e.g. eye-brow raising. Unfortunately, in telephone-based automated spoken dialogues, silences on the system’s part may not be so easily resolved if they are of the implicit kind, i.e. not verbally signalled - “Speak now”. In addition to turn-taking cues, there is also the implicit requirement for the signalling of the communication channel being open, i.e. making the user aware that the system has not crashed.

The work described in this paper focuses on the part of speech recognition dialogue where speech is captured and processed (by the recognition subsystem and the encapsulating application). Such a dialogue can be described as a three stage process. Firstly, a period where there is system output; in the form of a statement or question. This is followed by a period where the user speaks and then by a period of system processing. On completion of processing, the cycle repeats. Within the dialogue, it is essential that the user speaks at the appropriate time and that they are not encouraged to speak when the system is processing as no speech will be accepted.

A close analysis of current systems shows that there is no explicit representation of the different *states* of the recognition system (listening, processing[3]), though a tone or beep does provide an event representation[4] (listening

[1] This work has been carried out in the context of the ESRC-funded research project ‘Design guidelines for advanced voice dialogues’, under the Cognitive Engineering Programme, project no. L127251012.

[2] Now at Motorola Land Mobile Products, Viables Site, Basingstoke, UK.

[3] Some systems play music to the caller whilst the underlying system performing a lengthy operation. However, this is normally limited to telephony-based operations such as waiting for a telephone to be answered rather than application computational processing. Nevertheless, this interface representation was considered in the experimentation.

[4] The effectiveness of the representation is often subverted by a lengthy silence between the end of a prompt and the accompanying beep. Often this results in a stuttering affect where the user speaks just before the beep, hears the beep

starting event). This begs the questions, how does a new user know when to speak or know when not to speak? How does a user familiar with automated dialogues who knows when to speak know which prompts can be interrupted or more importantly when the system is processing the speech rather then listening? Finally, and arguably most fundamentally, how does any user know that the system has crashed?

This paper takes lessons from the visual and audio display literature and applies them to spoken dialogue design, with a focus on investigating the explicit auditory representation of a) the recogniser listening and b) the recogniser/application processing. By addressing these aspects, both novice and expert users can be supported in a more richly manifested telephone-based user interface.

2 What Should be Represented in the User Interface?

How does the paucity of representation effect the quality of interaction? To answer the question, it is helpful to ground the discussion in a wider interactional context. Brewster et al. [5] point out that users need to be able to interpret three key categories of system behaviour in order to interact effectively:

Events: Occurrences in the system domain that are system induced or user induced (via an input device). It is essential that the user is aware of some events, e.g. a system fault requiring immediate shutdown.
States: System variable values at a particular time. State knowledge is essential as it dictates which states may be next.
Modes: A particular mapping of user action to system action. Different modes allow the same user action to have a different effect. Examples are the `command` and `edit` modes in the UNIX Vi text editor.

Given these three categories problems occur when:

P1. Events/States/Modes not signalled to user (incorrect feedback)
P2. Events/States/Modes are rendered but not adequately
P3. The new state is not rendered correctly when an event causes a state change
P4. Events/States/Modes ambiguously rendered.

Given these problems, how do spoken dialogues fare? Table 1 examines different facets of spoken dialogues using Brewster's definitions. The following problems are evident in the renderings of spoken dialogues (P*n* refers to the generic problems identified earlier)
-All *events* are limited to a verbal rendering[5]. Such homogeneity may raise the *avoidability* of an event (P2).
-Most state renderings are not really renderings at all and are ambiguous (P1, P3, P4)
-The beep rendering is a transient representation of a potentially continuous state (P2)
-Mode renderings are not really renderings at all and are ambiguous (P1, P3, P4)

and stops and then speaks again; this clearly has negative repercussions for the recognition accuracy.
[5] Music on hold does provide a representation of the application processing state/mode.

Event	**Feedback/Rendering**
System utterance	*Spoken output*
User Utterance	*Spoken output*
Accepted Recognition	*Spoken output*
Recogniser error	*Spoken output*
Application error	*Spoken output*
State	
Recogniser state: listening	*Beep* *Silence*
Recogniser state: processing	*Silence*
Application state (ready for input, processing, other states ....)	*Silence*
Any application state	*Spoken output*
Current active vocab changed	*Nothing*
Word spotting active	*Nothing*
Talkover active (user can interrupt system)	*Nothing*
Mode	
What is the active vocab ?	*Nothing*
Talkoverable?	*Nothing*
Any application mode	*Spoken output*

Table 1: ESM Analysis of Speech Applications

The most obvious candidate for examination is the ubiquitous beep which is used to notify callers that it is their turn to speak. The beep can be examined as follows:

a) An *event* representation: recognition is starting
b) A *state* representation: a vocabulary is loaded and the recogniser is active,
c) A *mode* representation: the system will process and act on what is said by the caller when the utterance is completed[6]

In reality, only a **change** in the mode and state of the system is signalled by the *transient* beep. There is no explicit continuous representation of either the continuous mode or state. The situation is worse for the move from the recogniser 'listening' state to the application processing state/mode. Here, there is no representation of the fundamental modal change in the way that the user can interact with the system, i.e. their speech is no longer a catalyst, and yet a typical dialogue will be of the form below:

	<system> "Say a number between 0 and 8"
State/Mode 1: Listen	**Silence** <*speech recogniser listening*>
	<user>"Four"
State/Mode 2: Process	Silence <*speech recogniser processing*>
	Silence <*application processing*>
	<system> "Was that four?"
State/Mode 1:Listen	**Silence** <*speech recogniser listening*>

In State/Mode two, the system is deaf and the user is unable to alter the flow of the dialogue. Unfortunately, the

[6] A system-initiated event, i.e. the end-pointing algorithm postulates that the speech is at an end. However, as far as the caller is concerned they may not have finished speaking.

interface gives no signalling of this State/Mode.

Solutions in Visual Interfaces

Given the clear deficiency of avoiding explicit representations of the listening and processing states in spoken dialogues, it is helpful to look at how visual interfaces convey the same underlying *referent*. At this stage, the key difference of representation permanence must be stressed. Visual interfaces have the advantage of being able to represent the system state in a non-transient way; spoken dialogues on the other hand must rely on the ephemeral medium of, mostly verbal, sounds. Though continuous aural representations are possible, they may become irritating or intrusive. Secondly, visual interfaces provide a greater number of perceptual dimensions (size, shape, texture, luminosity, colour, etc.) for encoding.

The underlying application states identified in the previous section have the following analogues in visual interfaces:

Speech System	**Graphical System**
1 the recogniser waiting for an utterance to begin	the application is waiting for some input
2 end of utterance not detected	OK button not clicked
3 the application is processing	the application is processing

Table 2 Comparison of Aural and Visual Interfaces

How do visual interfaces deal with these situations? Case 1 is normally represented by blinking cursor or an input arrow. Case 2 is shown by a *modal* dialogue box which cannot be removed from the screen unless specified buttons are pressed. Case 3 relies on the cursor changing shape, e.g. an egg timer or watch. What all of these cases have in common is an attempt to show the *state* of the application in an intuitive iconic form. Assuming the user learns the mapping from icon to system referent then they will be able to interpret correctly the system state and act accordingly. The advantage of such forms is that they are small and can occur in parallel with the existing interface renderings, e.g. windows, tool bars, menu bars.

3 The Impact of Inadequate Representation on System Usability

Clearly, current spoken dialogues flagrantly fail to adequately represent a variety of important system modes, states and events; usability problems occur when a user acts incorrectly in response to the coarse interface abstraction. The wrong interpretation by the user can lead to disastrous consequences in the dialogue. An illustration is provided by Stifelman [13] who identified a problem in Apple's *VoiceNotes* prototype. In the VoiceNotes system, out of vocabulary recognition caused the system to remain silent and await a correct utterance. Stifelman notes:

"if the user spoke a command and VoiceNotes did not respond, rather than repeat the command, [as was expected], users waited for a response, thinking the system was still processing the input or busy performing the task" [Stilfelman, [13]: pp 184]

Further reactions prevalent in speech recognition systems are shown in Table 3.

User Task	**Reason for System Silence**	**User Response**	**System Effect**
Route call by saying a name	Prior to beep	Speak before beep	Misrecognition and confirmation loop
Say a command	Recogniser waiting for utterance to begin.	Silence	Silence error response
Say an amount of money	Recogniser processing an utterance	Give further information	Recognition of first amount only

Table 3: Possible Responses to Silence

A possible solution is to represent system state/mode/events using the same type of iconic forms prevalent in visual interfaces but limit them to the auditory channel.

3 Auditory Icons

Iconic Mappings

Work by Brewster et al. ([4],[5]) investigated an aural version of a scroll-bar (more detail of the sound mappings in the next section) vs. a normal scroll bar. The aural stimuli were arbitrary sequences of notes (earcons). Subjects performed search and navigate tasks within a graphical text editor; task completion time and cognitive workload[7] were measured. The results showed significantly faster operation with the auditory scroll bar in the navigate task where page boundaries had an explicit aural representation. Gaver [7] and Mynatt & Weber [10] used 'auditory icons', actual or stylised samples of real-world sounds, e.g. machine sounds, in a process control interface. Gaver emphasised the need to use sounds which were present in the real world since they allowed users to make typical interpretations. Informal analysis showed these sounds were good at representing parallel events. However, if continuous sounds were used to represent system state, e.g. Gaver used a continuous machine sound whose pitch was a function of the system's activity, they could be intrusive. Dutton et al. [6] used a mixture of auditory icons and more arbitrary sounds to represent a voicemail system's events and modes; there were no continuous sounds. Recall and preference was measured. Results showed that the more concrete and less arbitrary the icon the more correct icon associations that were made and recalled.

Within the spoken dialogue community, AlTech's [2] banking demonstrator uses a processing sound to identify that the recogniser was processing an utterance, a solution analogous to the visual egg timer. The WAXHOLM conversational system [3] uses meta-verbal utterances to represent different processing states, e.g. "Ummm" to signify the start of a long process and "Aha" to signify the system recognising a change in conversational topic. No formal investigation of the effectiveness of the sounds was carried out.

Dimensional Mappings

Normally auditory mappings and audio icons depict entities and actions, not values. However, there is also the opportunity to map quantitative aspects of domains onto aural dimensions. Aural representations can vary on a variety of dimensions. The variation is possible for both concrete representations such as Gaver's auditory icons [7] (these can vary on volume, pitch, speed) and Brewster et al's more abstract musical *earcons* (these can vary on dynamics, pitch, rhythm, timbre). Examples from the literature will now be described:

-Brewster et al. ([4],[5]) compared different *earcons* in a variety of visual interface widgets. Earcon sets varied on the quality of sound (synthesised instruments vs. pure sinusoids) in addition to the mappings are shown below.

Aural Dimension	**Referent Attribute**		
	Family, e.g. MS Word-related	Type, e.g. data, application	Same family/type
Pitch*			X
Rhythm		X	
Timbre	X		

-Alty et al. [1] used concrete aural representations which varied in pitch as a function of the domain value as part of a process control interface, the PROMISE system. The main finding was that it was important to be able to perceive gradations in the encoding sound.

-Rauterberg [12] used concrete representations in a process control system. The finding showed better error detection, i.e. hearing changes in system state, for the auditory condition.

7 Using the NASA Task Load IndeX (TLX) method; a set of multiattribute questionnaires.

-Mezrich et al. [9] mapped multiple time-series onto chromatic voices. The ability of subjects to correlate pairs of time-series data was measured for visual only and aural/visual displays. Aural representation was found to be better for short time sequences. Global pattern recognition was also better for the aural condition when all voices had similar timbre characteristics. Additionally, focus on a particular time-series could be provided by interactively making voices 'bright', i.e. adding more high frequency components.

-Pollack and Ficks [11] showed that a number of sound dimensions could effectively encode information in parallel. There results showed it was better to use more dimensions of the sounds than fewer dimensions with finer dimension gradations.

-Ludwig et al. [8] used *filtears*, signal processing of the auditory icons for different values of the same object, a menu sound could be aurally 'greyed out' by reducing the contribution of the sound's high frequency components.

In summary, the use of auditory icons is more effective where there are concrete *referents* since users can bring their world knowledge to bear on interpreting icons. Icons also provide a short-hand for system output which may reduce transaction time in spoken dialogues. Given the effectiveness of auditory icons, the question remains, which ones are suitable? In the case of the recognition phase of spoken dialogues, one referent (the listening mode/state) is an abstract concept, the second, processing, is a concrete one. There is no way to provide a concrete representation for an abstract concept which leaves the interface designer with metaphoric and abstract mappings. The abstract referent is an example of how qualities of the referent reduce the design space of the aural representation.

In general, key questions are:

-Do the sounds encourage the correct behaviour within a particular context?

-Does correct sound interpretation depend upon user familiarity with spoken dialogues?

It was thought that an experimental investigation would provide evidence of the effectiveness (or not) of non-verbal audio information in telephone-based spoken dialogues.

4 Experimental Design

To investigate the aural representation of the recogniser system state the experimental design was divided into two stages. A pilot phase focused on iconic sound mappings which compared the current implementation using beeps and silence to a metaphoric 'listening' sound and an abstract processing sound. Phase two was planned to use a larger subset of sounds This paper reports on the pilot phase.

The selection of test dialogue was seen as important as there may be a confounding effects on performance that are not connected with the chosen sounds, e.g. misleading prompts. There may also be domain dependent effects which will confound the result, e.g. a banking application may engender longer pauses before users speak, as well as cognitive complexity effects. Clearly, the experiment needed to make the aural representation the dominant independent measure. Therefore, the chosen dialogue required the caller to say a single digit between zero and nine. The system then provides confirmation of the recognised digit ("Was that x? Say yes or no"). For each of the two recognitions, a place-holder prompt simulates the application processing state; the result is then delivered. If the caller says that the recognised number is wrong the dialogue is repeated. Also, if the recogniser fails to obtain valid speech to process, e.g. the caller doesn't speak, the caller is asked to repeat the last utterance.

The first phase experiment aimed to gain initial responses to unfamiliar sounds in-dialogue. In this system, a male voice was used. So that interruption time could be measured, the system used talkover technology for all conditions (see independent variables below), including the 'speak after the beep' control condition. Therefore it was possible for the caller to successfully complete the dialogue even though they spoke before the beep. To ensure no learning effects, a between subjects design was used. Twenty-eight subjects were divided into four groups defined by the independent variables giving an unbalanced 2 x 2 between subjects design (see Table 4). The dependent measures where:

Interrupt time of listening state: Automatically measured relative to the start of the listening state, i.e. after the beep in the silence condition and at the end of the verbal part of the auditory icon condition. The value could be negative if subject spoke before the beep/end of the prompt and was the average of the two recognition states in the experimental dialogue.

Interrupt time of processing state: Automatic measure from the beginning of the processing state.

Errors: The number of loops for each stage in the dialogue, i.e. the digit capture loop, the yes no loop and the full application loop (the recognised number is disconfirmed). The former speech capture errors were the result of an internal recognition error, e.g. the caller did not speak loudly enough.

The independent variables were:

Subject Expertise: Two levels: novice and expert[8]

Auditory Type: Two levels: Silence (Beep-silence for listening state; silence for processing state), Auditory Icon (Sonar sound for listening state (**hear sound 'listen'**); stylised processing sound for processing state (**hear sound 'process'**)).

Subject Expertise	Listening Sound	Processing Sound
Novice	Silence (Beep)*	Silence*
Expert	Silence (Beep)*	Silence*
Novice	*Metaphoric*: **Sonar Ping**	*Concrete*: **Sci Fi Robot Processing**
Expert	*Metaphoric*: **Sonar Ping**	*Concrete*: **Sci Fi Robot Processing**

Table 4: Experimental Design (*=control)

5 Results

Interrupt Time for Listening State: The interrupt time variable was the average of both the 'Please say a number...' and '..yes or no" prompts, per subject. A two factor ANOVA was conducted on the variable. The main effects of Subject Expertise and Auditory Type were significant to $p<0.001$ [$F(1,27)=60.31$] and $p<0.05$ [$F(1,27)=3.83$]. The interaction effect (see Figure 1) between Subject Expertise and Auditory Type was significant to $p<0.001$ [$F(1,27)=9.89$].

Interrupt Time for Processing: Only two subjects interrupted the processing state. Both subjects were in the expert/silence condition.

Errors: See Table 5

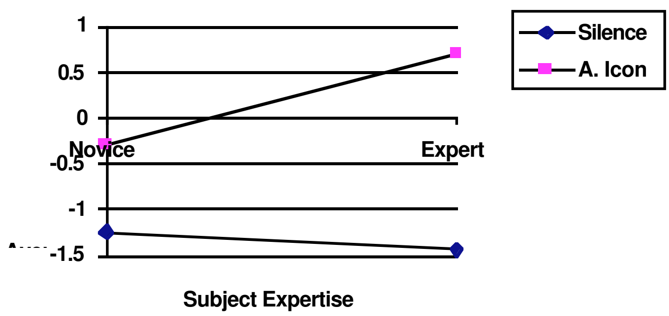

Figure 1 Interaction of Auditory Level and Subject Expertise Conditions for Average Interrupt Time

6 Discussion of Main Effects

Silence vs. Auditory Icons

Listening State Interrupt: Subjects in the silence condition tended to respond more quickly to prompts than their auditory counterparts. This is elucidated by examining the interaction effects which shows that in the

[8] Experts had extensive experience with spoken dialogue systems and were involved in system design and implementation. Novices had used dialogue systems but not as part of their day-to-day work.

silence condition expert subjects responded much more quickly than the novice subjects thus biasing the main effect result, an unsurprising result when one considers the familiarity of expert users with such dialogue systems. However, the auditory case shows a stronger effect in the opposite direction; experts tended to listen to the auditory icons because they were at odds with their extensive experience of spoken dialogue applications. It is also noticeable that on average the novice users spoke slightly before the auditory icon was heard which suggests that the auditory icons for the recogniser listening state was unnecessary since the imperative intonation of the system prompt induced the correct behaviour regardless.

Processing State Interrupt: No processing interruptions occurred in the auditory icon treatment, thus enhancing the usability of the system by avoiding the potential drawback of using silence for the processing state.

Errors: Fewer errors were made by novice and expert users in the auditory icon condition. For Novice users the error result can be attributed to the auditory icon in the second system request, “Was that X yes or no?”, where unlike the first request, subjects did not interrupt and heard the icon. This result is masked due to the *average* interruption time for these two prompts being measured.

Novice vs. Expert

Listening State Interrupt: Novice subjects were significantly quicker to respond to prompts ($p<0.001$); often speaking before the beep in the silence treatment. Given that like the experimental system, commercial systems often leave a second between the end of a prompt and the beep. This highlights a major usability problem since in a non-talkover system the subjects speech would not be recognised. This result corroborates the first example in Table 3.

Processing State Interrupt: The processing interruptions occurred in the expert treatment. This may show how expert users were more impatient than their novice counterparts.

Errors: Overall, novices made more errors than experts.

Novice/Silence	Expert/Silence	Novice/A. Icon	Expert/A. Icon
3,3,1	5,1,1	1,1,1	2,0,0

Table 5 Digit, Y/N and Application Errors

7 Conclusion and Further Work

In summary, the pilot experiment showed auditory icons need to be carefully chosen for the particular referent under consideration and the typical user experience. Overall, icons caused fewer errors than the control condition for all users. The experience of callers with spoken dialogue systems had a strong effect on the success of auditory icon deployment. Expert users listened to ‘recogniser listening’ icons rather than spoke which is in stark contrast to their speed of response in the control condition, perhaps suggesting that a period of acclimatisation is required. For novice users, icons may not be required where the imperative intention of the system is made clear as users were quite willing to speak. A potential caveat is that the system must use talkover to cater for users speaking immediately (or slightly before) the end of a prompt. There was some indication that in less well signalled systems with states such as processing, an auditory icon will deter both novice and expert users from making futile verbalisations.

The next phase of experimentation will examine the use of sound ecologies which render a wider range of system states and events, e.g. particular recogniser errors, talk-over and confidence measure values, as well as application events, e.g. a new message has arrived, an operator is unavailable (as in [6], [7]). Of particular interest is the use of event representations to replace verbal prompts thus shortening transaction time for repetitive tasks.

8 References

[1] Alty, J. L.; M. Bergan, P. Craufurd, Experiments Using Multimedia in Process Control: Some Initial Results. Computer & Graphics, Vol. 17 (3), 1993, pp. 205-218.

[2] Applied Language Technologies, http://www.altech.com/

[3] Blomberg, M, An Experimental Dialogue System: Waxholm, In: Proceedings of Eurospeech '93, Berlin, 1993, pp 1867-1870.

[4] Brewster , S A., Wright P C, Dix, A J, Edwards A D N , The Sonic Enhancement of Graphical Buttons, In: Proc. of Interact'95, Amsterdam: April, New York: ACM Press, 1994, pp 43-48.

[5] Brewster , S A., Wright P C, Edwards, A D N, The Design and Evaluation of an Auditory-Enhanced Scroll Bar In: Proc. of ACM SIGCHI'94, Boston: April, New York: ACM Press, 1994, pp 173-179.

[6] Dutton, D, Kamm, C, Boyce, B, Recall Memory for Earcons, In: Proc. of EuroSpeech, Rhodes, August., 1997

[7] Gaver, W, The SonicFinder: An Interface that Uses Auditory Icons, In Journal of HCI, Vol. 4, 1989, pp 67-94.

[8] Ludwig, L. F., N. Pincever, M. Cohen, "Extending the Notion of a Window System to Audio". IEEE Computer, August, 1990, pp 66-72

[9] Mezrich, J. J., S. P. Frysinger, R. Livjanovski, Dynamic Representation of Multivariate Time-series Data Journal of American Statistical Association, Vol. 79, 1988, pp 34-40.

[10] Mynatt, E. D, Weber G, Nonvisual Presentation of Graphical User Interfaces: Contrasting Two Approaches, In: Proc. of ACM SIGCHI'94, Boston: April, New York: ACM Press, 1994, pp 166-172.

[11] Pollack I, L. Ficks, Information of Elementary Multidimensional Displays. In Journal of Acoustical Society of America, Vol. 26, 1954, pp 155-158.

[12] Rauterburg, M. About the Importance of Auditory Alarms During the Operation of a Plant Simulator. In 'Interacting with Computers' Vol. 10, 1998, pp 31-44.

[13] Stifelman et al., VoiceNotes, In: Proc. of INTERCHI'95, Amsterdam: April, New York: ACM Press, 1993