*Int. J. Man-Machine Studies* (1977) **9**, 449–464

# The Telephone Enquiry Service: a man–machine system using synthetic speech

I. H. Witten and P. H. C. Madams
*Man–Machine Laboratory, Department of Electrical Engineering Science, University of Essex, U.K.*

(*Received 7 October 1976 and in revised form 23 February 1977*)

The Telephone Enquiry Service is a computer system which allows interactive information retrieval from an ordinary touch-tone telephone. For input, the caller employs the touch-tone keypad, and the computer replies with a synthetic voice response. The service has been in fairly continuous operation for around one year, using a small time-shared computer in conjunction with an internal 200-line telephone exchange, and has been widely used by people with no special interest in synthetic speech.

An unusual feature of the system is that the speech is generated by rule from a phonetic representation. A satellite computer, acting as a peripheral to the main machine, performs this task in real time, and controls the parameters of an analogue speech synthesizer. This constitutes an extremely economical and flexible method of speech storage, whose only real disadvantage is the low quality of articulation of the output. A major conclusion of the work is that even low-quality speech is acceptable to casual users, if the service is sufficiently interesting and useful to them.

## 1. Introduction

Interactive computers are being used more and more by untrained people without much experience of them. As usage grows and processing costs continue to decline, the provision of terminals and distribution equipment is increasingly tending to dominate the cost of computer systems. The major man–computer interfaces in widespread use are typewriters and visual display units. Although some lower cost terminals with simple keyboards exist, these generally suffer from limited displays. Unfortunately, a restricted output channel from the computer inhibits man–machine communication more than almost anything else—even the ordinary teletype is too slow for comfort, although it operates far faster than most typists.

Analysis of potential remote information-retrieval applications shows that there is a requirement for inexpensive peripherals with elementary input devices but sophisticated output facilities, so that the user's conversation with the computer is minimal but the feedback from it is maximal. Speech output from the computer can provide this versatile feedback at very low cost in distribution and terminal equipment. However, digitally-encoded speech in conventional forms is expensive to store and cumbersome to manipulate—as well as being rather tedious to gather. Speech synthesized by rule from a textual or phonetic input appears to offer the promise of economical, flexible speech output, occupying little more store than messages intended for typewriter-like devices.

The outstanding disadvantage of synthetic-speech-by-rule output is that it is usually quite difficult to understand, especially in large quantities. Although it is continually being improved, we felt after several years' experience with synthetic speech that the most dramatic enhancement in intelligibility would result from the conversation being

given a proper context and purpose, and began to implement a Telephone Enquiry Service to test artificial speech in a complete system (Corbett, 1974).

After a brief review of speech synthesis methods, this paper describes the use of the service and how it appears to the caller. The chief design problem was the implementation of a man–machine interface which utilized the disparate input and output channels effectively, and so special considerations relating to the auditory display in conjunction with keypad entry are treated next. The system is designed to facilitate creation of new services by programmers without special knowledge of speech, and an appropriate application programming environment, embedded in BASYS (a considerable extension and rationalization of Dartmouth Basic), was developed. This is discussed in section 5. The subsequent section returns to the problem of speech synthesis, and details are given of how utterances are represented within the system, together with an overview of the speech synthesis techniques employed. Finally, user reactions to voice response are described, and these lead to some suggestions for its effective use in future systems.

## 2. The synthesis of speech

Although voice output from computers has been available commercially for several years, most existing systems employ methods which involve recording and replaying humanly-spoken utterances. Clearly, a tape recorder with an auxiliary addressing mechanism will suffice to generate a limited number of speech messages, and could be used in an automatic voice response system if only it were fast and reliable enough. The IBM 7770 Audio Response Unit employs magnetic drums, each rotating twice a second and able to store up to 32 500-millisecond words which can be accessed randomly. The Cognitronics Speechmaker has a similar structure, but with the analogue speech waveform recorded on photographic film. Although one can arrange to store longer words by allowing overflow on to an adjacent track at the end of the rotation period, the discrete time-slots provided by these systems make it virtually impossible for them to generate connected utterances by assembling appropriate words from the store.

A more flexible voice response facility is obtained if the audio speech signal can be encoded digitally in a way which is efficient enough to make it feasible to store the information on an ordinary computer bulk storage device. Direct digital encoding of telephone-quality speech using ordinary pulse-code modulation produces about 50,000 bits of information for each second of speech (6 bits per sample with an 8 kHz sampling rate), which renders storage costs rather expensive—but rapidly becoming less so—for any but the smallest amounts of speech. However, the major drawback to direct storage of the waveform is that one cannot construct natural-sounding utterances by concatenating individual words which were recorded in isolation, or in a different context (Stowe & Hampton, 1961). The glue which is needed to stick together words of speech is a mysterious adhesive which causes quite radical changes in the adherents.

Perhaps the most perceptually significant contextual effect which must be taken into account when forming connected speech out of isolated words is pitch. The intonation of an utterance, which manifests itself as a continually changing pitch, is a holistic property of the utterance and not the sum of components determined by the individual words alone. Happily, and quite coincidentally, communications engineers in their quest for reduced-bandwidth telephony have invented methods of coding speech that

separate out the pitch information from that carried by the articulation (Dudley, 1939). These techniques of *channel vocoding* are quite suitable for use in computer voice response systems. This can reduce the digital information rate to as low as 1000 bits per second of speech, and allow one to alter the intonation of each word to conform to a uniform contour for the complete utterance. The IBM 7772 Audio Response Unit is an example of successful commercial application of these techniques (Buron, 1968).

Unfortunately, the glue's effect is rather too far-reaching to allow re-synthesis of words recorded in different contexts to produce natural-sounding connected utterances, even with the sort of manipulation of pitch that a vocoder-based system allows. The rhythm of speech plays an important part in its naturalness, and adjustment of the timing of pre-recorded, vocoded words to produce a flowing rhythm is not really feasible. Furthermore, there are important contextual effects due to articulation—we do not really leave gaps between words when we speak!—which cannot be simulated from recordings of isolated words.

The Telephone Enquiry Service which is the subject of this paper uses a hardware formant-based speech synthesizer. Although most such devices, including the one used in the present system, have only a single-channel output capability, multi-channel ones are currently under development commercially (Underwood & Martin, 1976), and so extension of the Service to accommodate several lines would not present any new technical problems. It is possible to drive a formant synthesizer with an encoded version of a natural utterance—although the coded representation, unlike that for the channel vocoder at the heart of the IBM 7772, is not easy to elicit *automatically* from natural speech. Nevertheless, if enough care is taken preparing the parameters, such a synthesizer can reproduce utterances so faithfully that they are indistinguishable from the original, even under high-fidelity listening conditions (Holmes, 1973). Instead of imitating natural utterances, however, the Telephone Enquiry Service stores speech as text in a phonetic form, and computer software, working in real time, converts this representation into control parameters for the synthesizer. The speech is completely synthetic and produced without any reference to particular natural utterances (although the synthesis program, of course, incorporates a great deal of general information about speech). This means that problems of out-of-context words do not arise. It also provides a convenient and extremely economical representation for storage—a phonetic transcription occupies only slightly more store than ordinary printed English, of the order of 50 bits per second of speech (compare with 50,000 bits/second for a digitized waveform!).

Apart from the phenomenal storage efficiency, the advantages of storing phonetics and synthesizing speech in real time, over the more conventional methods of handling encoded versions of natural speech, are quite striking, especially from the point of view of system development and maintainance. The utterances can be stored as phonetic text, incorporated into the particular parts of the program where they are used. Thus, given adequate software and hardware organization, the output requires a simple "speak" command which takes a short character-string as argument—exactly the same as teletype output. (However, we hope to demonstrate convincingly in the following pages that specific non-teletype programming techniques are important for comfortable interaction.) No special mechanisms for entering speech are needed. To add new utterances to a voice response system using stored human speech, one must assemble together special input hardware; a quiet room; a particular person, so that the system has a consistent voice; and (probably) a dedicated computer. This discourages the application programmer

from making cut-and-try attempts to render the man–machine dialogue as natural as possible in the final stages of debugging. The synthesis-from-phonetics technique means that he can change a speech prompt as simply as he could a teletype prompt, and evaluate its effect immediately.

A further potential advantage of this method of utterance storage is that the prosodic features of an utterance, and in particular its stress, can be altered by the program without any need to store different versions of it. This facility has not been used in the present system.

By far the greatest drawback to synthetic speech in voice response systems is the low quality of articulation. Fortunately, this is continually being improved as research by acoustic phoneticians uncovers new facts about the structure of speech, and although at present crude methods of concatenating pre-recorded words may well produce speech which is more intelligible, synthesis-by-rule's fundamental advantage of greater control over suprasegmental features will render it superior in quality in the future, especially in systems which need to generate a great variety of utterances. A lesser inconvenience is the necessity for application programmers to acquire skill in phonetic transcription. Fortunately, an interactive situation where the effects of modifications to the transcription can be heard immediately provides an ideal environment in which to learn this art. Indeed, techniques of automatic phonetic transcription from English are beginning to show promise (Witten, 1976, describes a system with a measured reading age of about 12 years according to a standard test), and quite a significant rate of error is tolerable if immediate audio feedback of the results is available so that the operator can adjust his text input to suit the pronunciation idiosyncrasies of the program.

The method of phonetic representation in the current system is described in a subsequent section. First, let us see how the interaction appears to the telephone caller.

## 3. The Telephone Enquiry Service

The Telephone Enquiry Service is a computer system which allows interactive information retrieval from an ordinary touch-tone telephone. For input, the caller employs the touch-tone keypad illustrated in Fig. 1, and the computer generates a synthetic voice response. The process of making contact with the system is as follows:

| | |
|---|---|
| CALLER: | Dials the service. |
| COMPUTER: | Answers telephone. |
| | "Hello, Telephone Enquiry Service. Please enter your user number." |
| CALLER: | Enters user number. |
| COMPUTER: | "Please enter your password." |
| CALLER: | Enters password. |
| COMPUTER: | Checks validity of password. |
| | If invalid, the user is asked to re-enter his user number. |
| | Else, |
| | "Which service do you require?" |
| CALLER: | Enters service number. |

Advantage is taken of the disparate speeds of input (keyboard) and output (speech) to hasten the dialogue by imposing a question–answer structure on it, with the computer taking the initiative. The machine can afford to be slightly verbose if by so doing it makes the caller's response easier, and therefore more rapid. Moreover, operators who

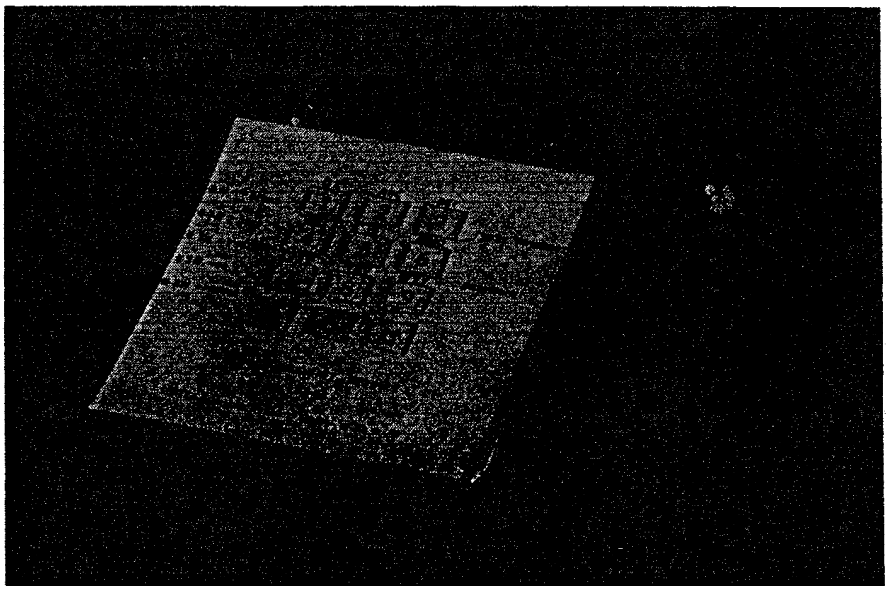

FIG. 1. The touch-tone telephone terminal.

are experienced enough with the system to anticipate questions can easily forestall them by typing ahead (see next section).

A prime aim of the system is to allow application programmers with no special knowledge of speech to write independent services for it. Here is an example of the use of one such application program, the Stores Information Service (full details can be found in Corbett, 1974).

COMPUTER: "Stores Information Service. Please enter component name."
CALLER: Enters "SN7406#" (entry of alphabetic characters is described in the next section).
COMPUTER: "The component name is SN7406. Is this correct?"
CALLER: Enters "*1#" (system convention for "yes").
COMPUTER: "This component is in stores."
CALLER: Enters "*7#" (command for "price").
COMPUTER: "The component price is 35 pence."
CALLER: Enters "*8#" (command for "minimum number").
COMPUTER: "The minimum number of this component kept in stores is 10."
CALLER: Enters "SN7417#".
COMPUTER: "The component name is SN7417. Is this correct?"
CALLER: Enters "*1#".
COMPUTER: "This component is not in stores."
CALLER: Enters "*9#" (command for "delivery time").
COMPUTER: "The expected delivery time is 14 days."
CALLER: Enters "SN74099#".
COMPUTER: "Component name not known."
CALLER: Enters "SN74099#".

COMPUTER: "No information available."
CALLER: Enters "*0#".
COMPUTER: "Which service do you require?"

A distinction is drawn throughout the system between *data entries* and *commands*, the latter being prefixed by a "*". In this example, the programmer chose to define a command for each possible question about a component, so that a new component name can be entered at any time without ambiguity. The price paid for the resulting brevity of dialogue is the burden of memorizing the meaning of the commands. This is an inherent disadvantage of a one-dimensional auditory display over the more conventional graphical output: presenting menus is tedious and long-winded. In practice, however, for a simple task such as the Stores Information Service it is quite convenient for the caller to search for the appropriate command by trying out all possibilities—there are only a few.

1 — Tells the time
2 — Biffo (a game of NIM)
3 — MOO (a game similar to that marketed under the name "Mastermind")
4 — Error demonstration
5 — Speak a file in phonetic format
6 — Listening test
7 — Music (allows one to enter a tune and play it)
8 — Gives the date

101 — Stores information service
102 — Computes means and standard deviations
103 — TESS telephone directory

411 — User information
412 — Change password
413 — Gripe (permits feedback on services from caller)

600 — First year laboratory mark entering service

910 — Repeat utterance (allows testing of system)
911 — Speak utterance (allows testing of system)
912 — Enable/disable user 100
913 — Set/reset "DECtape mounted" flag
914 — Set/reset "Demonstration mode" (prohibits access by low-priority users)
915 — Inhibit games
916 — Inhibit the MOO game
917 — Inhibit ** access (if ** access is enabled, users may log in without having to enter their user number and password)

FIG. 2. Summary of services currently available.

The problem of memorizing commands is alleviated by establishing some system-wide conventions. Each input is terminated by a "#". The following commands always have the same meaning:

| | | |
|---|---|---|
| *# | — | Erase this input line, regardless of what has been typed before the "*". |
| *0# | — | Stop. Used to exit from any service. |
| *1# | — | Yes. |
| *2# | — | No. |
| *3# | — | Repeat question or summarize state of current transaction. |
| # alone | — | Short form of repeat. Repeats or summarizes in an abbreviated fashion. |

A summary of services currently available on the system is given in Fig. 2. A priority structure is imposed upon them, with higher service numbers being available only to higher priority users. Services in the lowest range (1–99) can be obtained by all, while those in the highest range (900–999) are maintainance services, available only to the system designers. Note that access to the lower-numbered "games" services can be inhibited by a priority user—this was found necessary to prevent over-use of the system (see section 7). Another advantage of telephone access to an information retrieval system is that some day-to-day maintainance can be done remotely, from the office telephone.

## 4. Auditory display and keypad entry

At the outset, it was decided to use a small satellite computer as an input-output processor for the main system. The generation of speech from a phonetic representation is a fairly complex and time-consuming procedure, and the use of a dedicated machine for synthesis relieves the load from the main processor, allowing it to treat the peripheral much as it would a full-duplex teletype. The telephone is connected to the satellite computer, so the routine business of detecting a call, waiting for a few rings, answering, connecting the line, hanging up, and disconnecting, is transparent to the main machine. The touch-tone keypushes are detected and converted into ASCII codes by the peripheral computer. Fig. 3 shows a block diagram of the system hardware.

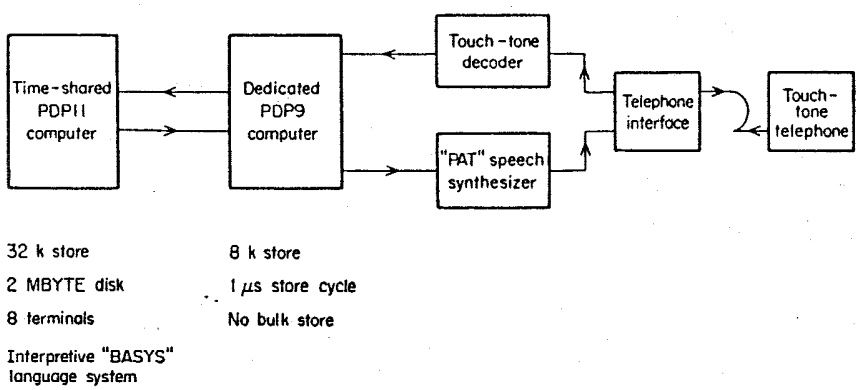


FIG. 3. Block diagram of the Telephone Enquiry Service hardware.

Originally, we thought that close attention would have to be paid to the echoing of messages keyed in by the caller. People make quite a high proportion of mistakes using the keypad, especially with alphanumeric data, and it was felt that a positive response like an echo would be necessary for confirmation. In practice, however, the simple command structure means that a positive response—namely, the result of the transaction —*does* follow every input command almost immediately, without the need for explicit echoing. Thus if an erroneous entry occurs, the system will reply right away with a response to the wrong command, and the user, realizing his mistake, will re-enter his request. Of course, it is important that the response to a command always identifies the command itself. However, this is easy to do in quite a natural way, and increases the system's verbosity only slightly.

It may happen that certain user commands call for a large amount of processing. If so, it is the application programmer's responsibility to ensure that, when the command is accepted, an appropriate response is generated immediately to confirm that it has been recognized.

Because the input/output system is duplex and fully buffered, it is possible to "type ahead". Experienced users often take advantage of this, for example, to enter their user number, password, and service number as a single keying sequence. When this happens, there is no need for auditory prompts, and the system invariably inspects the input buffer before generating a prompt, to ensure that it really is necessary. As one becomes familiar with the service, one quickly and easily learns to forestall expected prompts by typing ahead. This provides a very natural way for the system to adapt itself automatically to the experience of the caller.

A facility is also provided to allow interruption of the output. Although the operating system under which the Telephone Enquiry Service runs does not support a software interrupt, the input buffer is frequently polled to check for a "stop" command ("*0#"). If one is found, the service is left and the system returns to the "Which service do you require?" state (see example above).

A strong restriction on the input terminal would seem to be its purely numeric, 12-key nature. In fact, this proves to be hardly a restriction at all, for the kind of command and simple data entry that we employ. Two or three letters are associated with each digit, in a manner which is fairly standard in touch-tone telephone applications (Kramer, 1970; Newhouse & Sibley, 1967). These are printed on a card overlay that fits the keypad (see Fig. 1). Although true alphanumeric data entry would require a double or triple key press for each character (Kramer, 1970, describes experiments with two different methods), the ambiguity inherent in a single-key-per-character convention can usually be resolved by the computer, if it has a list of permissible entries. For example, the component names SN7406 and ZTX300 are read by the machine as "767406" and "189300", respectively. Confusion rarely occurs if the machine is expecting a valid component code. The same holds true for people's names, and for file names—although with these one must be careful not to identify a series of files by similar names like TX38A, TX38B, TX38C. It is simple for the machine to detect the rare cases when ambiguity exists, and respond by requesting further information: "The component name is SN7406. Is this correct?" (In fact, the Stores Information Service illustrated in the dialogue above is defective in that it *always* requests confirmation of an entry, even when no ambiguity exists.)

## 5. Features of the application programming system

The Telephone Enquiry Service is implemented in BASYS, an interpretive language developed specifically for interactive non-numeric man–machine systems (Gaines & Facey, 1975). BASYS is modelled loosely on Dartmouth Basic, with the addition of SNOBOL-like pattern matching and string decomposition capabilities and powerful input–output facilities, which are closely matched to the operating system calls provided by the host computer system.

The software comprises a suite of overlays, one (or more) for each service program. To make life as easy as possible for the application programmer, a central kernel of code —the system monitor—is retained with his program all the time his service is being executed, and can be called upon to perform various system functions. Each service program resides in an individual file, and is called in whenever that service is requested.

During development, the programmer is at liberty to rename the file, making it inaccessible to the system. Any user requests for the service will then result in a "Service not available" message (which is different from the "No such service" response which is generated when a non-existent service is solicited). Thus development can proceed while the system is running, and since only a single telephone line is accomodated, the programmer can call the system before temporarily renaming his program file with the correct name, ensuring that he alone has access to it in its undebugged state. (This makes a virtue out of necessity. However, other mechanisms—for example the priority structure —are available to restrict access to a service temporarily, and would be used while debugging in a multi-line system.)

Since no mechanism for defining new commands is available in BASYS, the system functions are performed by subroutine calls to the above-mentioned core-resident system monitor. The kind of functions provided for output are:

(a) synthesize an utterance specified in phonetic form, passed as a character string argument;
(b) synthesize an integer number, passed as an integer expression;
(c) synthesize a string of alphabetic characters and digits, passed as a character string.

In order to implement the last two functions, the system contains an internal alphanumeric-to-phonetic pronunciation table. The second function constitutes an algorithm for converting an integer into an appropriate utterance, for example 121—> "one hundred and twenty-one"; 1119—> "one thousand one hundred and nineteen". This procedure is not difficult, but requires some attention to detail, particularly with regard to treatment of embedded zeros and the word "and". The third function provides the capability to output file names and other codes (such as the "SN7406" of the earlier Stores Information Service example). It is also useful for synthesizing decimal numbers, and the service programmer has the choice of, for example, "eight point zero one" and "eight point oh one" by re-coding the character string which represents the number.

In all cases of output, a message is not actually spoken until a terminator character is sent. This is analogous to many teletype–oriented operating systems, which do not print partial lines but wait until a carriage return or other similar special character is output. It is essential in a speech system, to eliminate any real-time problems which may result in gaps in the speech when pieces of a message are generated by different parts of the program. An example is "The component name is SN7406", where the initial context is determined separately from the specific component code. The system collects the utterance in a buffer, and the terminator initiates synthesis. Different terminators, for example "." and "?", could be used to influence the intonation of the utterance (although at present, as explained in the next section, intonation is determined by a code at the beginning of the utterance).

The core-resident part of the system software also incorporates subroutines for input buffering and decoding. Again, these are designed to encourage service programmers to implement well-thought-out dialogues with as little trouble as possible.

The ASCII characters corresponding to the caller's key-pushes are buffered, along with special messages from the satellite processor like "hang up" and "new call". The system monitor, whenever it is entered, scans the buffer backwards for "hang up" and "*#" (the "stop" command). If either is found, all inputs prior to it are removed from the buffer, and the system exists from the service program, in one case to the quiescent state, and in the other to the "Which service do you require?" state.

Functions are available that allow the service programs to receive the ASCII characters directly as they are input, or to receive them after they have been processed, that is decoded either as commands or as integer data entries. Although commands are supposed to begin with "*", the decoding mechanism will map a plain integer on to a command as though it were preceded by "*". This means that at any point in a service where *only* a command is expected, the corresponding integer alone will do.

```
09:44:21  TELSER - V4B     6-12-74
09:45:24  LINK RE-CONNECTED
09:45:26  NEW CALL
09:45:30  HANG UP
09:45:30  ***ERROR TRAP LINE 3009
09:45:30  OVERLAY LOADED AT  768
09:45:32  HANG UP
09:45:43  NEW CALL
09:45:47  USER - 123     A.CORBETT
09:46:06  SERVICE 411 - USER INFORMATION
09:46:16  SERVICE EXIT
09:46:20  SERVICE 400 - CHANGE PASSWORD
09:46:34  PASSWORD CHANGED TO 2659
09:46:34  SERVICE EXIT
09:46:39  SERVICE 1 - TIME
09:46:51  SERVICE EXIT
09:46:55  SERVICE 4 - ERROR DEMONSTRATION
09:47:05  ***ERROR TRAP 5004
09:47:14  HANG UP
09:47:16  OVERLAY LOADED AT  768
09:47:22  NEW CALL
09:47:27  USER  - 100   USER.100
09:47:36  SERVICE 101 - STORES INFORMATION SERVICE/VERSION 2
09:47:38    SN7400
09:47:49  PRICE IS   15
09:47:58  MINIMUM NUMBER IN STORES IS   25
09:48:00  DELIVERY TIME IS  28
09:48:22    ZTX300
09:48:36  MINIMUM NUMBER IN STORES IS 200
09:48:38  DELIVERY TIME IS   7
09:48:45  PRICE IS  10
09:48:52  SERVICE EXIT
09:48:59  HANG UP
09:49:00  OVERLAY LOADED AT  768
```

FIG. 4. Extract from the system log.

Automatic handling of either or both of the "repeat"commands—"#" and "*3#"—on the basis of regenerating the preceding prompt or prompts, was considered but rejected because it constrains the application programmer irrevocably. Subsequent experience has shown that sophisticated services have "repeat" procedures much more complicated than merely re-synthesizing the last utterance.

A further feature of the system monitor is error recovery. A log is maintained automatically of new calls, user identities, service programs used, and internal errors. A monitor subroutine is available as an error exit from service programs—a message is printed in the log and the dialogue reverts to the "Which service do you require?" state. More important, for a system which is intended to enable service program development while it remains available to users, is the monitor's ability to intercept language errors in service programs, such as "unknown command" or "illegal expression". This relies on a feature of BASYS that allows a program to trap system errors and take appropriate recovery action itself. An occurrence of a language error is logged automatically, the

telephone caller is informed, and the system reverts to the "Which service do you require?" state. The service programmer can examine the log and correct the error at his leisure; in the meantime the system continues to be available.

An extract from the log is shown in Fig. 4. Error traps are asterisked; the integer given with the error is the BASYS line number where it occurred.

## 6. Storage and generation of speech

The first computer programs for synthesis of speech by rule from a phonetic representation were developed by Kelly & Gerstman (1961) and Holmes *et al.* (1964). These early systems accepted strings of phonemes with additional "modifier" elements, the most important of which controlled the duration of the phoneme segments (determining the rhythm of the utterance), and the pitch movements (determining its intonation contour). Other modifiers allowed "tweaking" of the output to enhance the intelligibility of the speech. Preparing input for such systems is a difficult and skilful task, requiring considerable linguistic training and much experimentation.

Phoneticians divide features of natural speech into segmental ones (giving segments of syllables) and suprasegmental or prosodic ones (relating to properties of speech other than pure articulation), and this distinction can be applied to synthetic speech as well. The use of modifier elements to specify rhythm and intonation means that the suprasegmental features are put in by hand, and only the segmental ones are synthesized "by rule".

However, if synthetic speech is to be used as a computer output medium rather than as an experimental tool for linguistic research, it is important that the method of specifying utterances is natural and easy to learn. Suprasegmental features must be communicated to the computer in a manner somewhat simpler than individual duration and pitch specifications for the phonemes which constitute the segmental description of an utterance. Halliday (1967) has developed a standard notation for conveying some of the prosodic features of utterances, as a by-product of his main goal of classifying the intonation contours of English. He has used his system of notation and classification to help foreigners speak English (Halliday, 1970)—and this emphasizes the fact that it was designed for use by laymen, not just linguists.

Here is an example which illustrates Halliday's notation.

//4 ^ but the / <u>can</u>didates / don't get nine / grades //

Three levels of stress are distinguished: the tonic or major stress (marked by____), the foot stress (marked by /), and unstressed syllables. An intonation contour is specified by a code (<u>4</u> in this case) at the beginning. The double slashes delimit the domain of the intonation contour: an utterance may consist of many of these "tone groups". The symbol "^" indicates a silent beat, which is a non-final pause.

We communicate utterances to the computer using a scheme based on Halliday's, but with a phonetic transcription:

6 ^ BAT DHUH /*KAANDIDITS
/DUHUNT GNAAET NAAIN /GREIDS.

The machine-readable transcriptions of International Phonetic Alphabet symbols are straightforward. We retain the foot stress marks (/) and use * for tonic stress, but employ

normal punctuation marks (., !, ?, -), rather than //'s, to delimit tone groups. Our method of intonation specification, whilst originally based on Halliday's, is intended to be more flexible, and the code at the beginning of an utterance does not necessarily indicate one of Halliday's tone groups.

The speech synthesis software operates in two stages. Firstly, the suprasegmental properties of rhythm and intonation are analyzed, the result being an assignation of duration to each phoneme, with pitches specified at various points throughout the utterance. Secondly, the segmental synthesis is performed, by interpolating each synthesizer parameter between set points specified for each phoneme, and letting the parameter values linger at the set point for part of the duration of the phoneme. (In practice, the segmental synthesis procedure is rather more complicated than this.) However, we are increasingly inclined to regard the division between the two phases of synthesis as artificial, and future versions of the software will reflect this by preserving all information from the prosodic analysis for possible use by the segmental synthesis part.

Let us illustrate some of the considerations that are taken into account by the prosodic routines (full details can be found in Witten, 1977). The feet in the pretonic—up to the "/*" marker—are viewed as forming a succession of minor stress points, with a certain syllabic rhythm between each, and with a regularly recurring pattern of intonation (on which may be superimposed a steady rise or fall). This intonation pattern changes abruptly at the tonic stress point, either by an unexpected (or, at least, unheralded) pitch discontinuity, or by a reversal in direction of pitch movement; and the change is perceived as tonic stress. In contrast with the pretonic, the overall intonation pattern on the tonic flows smoothly, without taking account of feet, leaving only the temporal rhythm of the syllables to indicate the minor stresses at foot boundaries.

Our starting point for imitating the rhythm of natural English speech is the theory of isochrony of feet (Abercrombie, 1964). Although the question of isochrony has long been debated, there seems to be general agreement that there is at least a tendency for foot boundaries to be equally spaced in time. Furthermore, taking the foot as the foundation for the rhythm of synthetic speech avoids the considerable difficulties of those who treat duration as a *segmental* phenomenon (see, for example, Umeda, 1976).

In order to apportion the foot duration between the individual phoneme segments, each foot is split into its constituent syllables, and from their types a rhythm for the foot is determined. The foot duration is divided amongst the syllables according to the rhythm. The question of where each syllable begins and ends is a difficult one: research indicates the existence of a rhythmic "tapping point" in a well-defined place (usually just before the vowel). The phonemes of a syllable are then classified into those whose duration is intrinsic and those for which it is extrinsically determined by the rhythm, and the syllable time is divided amongst them accordingly (Lawrence, 1974).

For the Telephone Enquiry Service, an early implementation of the speech synthesis system was employed in which the two stages of prosodic and segmental synthesis were rigidly separated. Moreover (and regrettably), the suprasegmental part was performed by an off-line procedure, and only the segmental synthesis was done in real time by the satellite computer mentioned earlier. The form of the intermediate output is shown in Fig. 5. The duration of each phoneme is specified, preceded by the marker "D"; and the pitch is given at various points through the utterance with a "P" marker, followed by the time at which the pitch is specified (after the notional beginning of the current phoneme) and its value. All durations are in units of 10 milliseconds. The pitch is

interpolated linearly between the specification points by the segmental synthesis software.

The fact that the intermediate form of phonetic representation of Fig. 5 was stored in service programs meant that full advantage could not be taken of some of the features of synthetic-speech-by-rule output. It ruled out dynamic adjustment of intonation according to the demands of the interactive situation, which could probably have been used effectively, particularly when replying to "repeat" requests, when increased emphasis would be especially useful. The rate of the speech could even have been adjusted in accordance with the caller's speed of keying, which presumably corresponds to his experience and facility with the service! A future version of the Telephone Enquiry Service is planned to investigate these effects.

```
^    D16  P1 31  P13 29
DH   D8
UH   D10
D    D6
AA   D8   P0 33
U    D7
N    D5   P1 31
B    D4
EE   D11
T    D9
M    D4
AR   D9    P0 34
K    D8
S    D4   P1 32
DH   D5
UH   D9
B    D5
UH   D9
G    D5
I    D9   P0 36
N    D5
I    D10
NG   D14   P9 31
UH   D6
V    D4
DH   D4
UH   D9
B    D5
AR   D16   P0 31
```

FIG. 5. Output of the suprasegmental synthesis routine.

## 7. User reactions to voice response

One of the most surprising outcomes of the experimental Telephone Enquiry Service is that people are prepared to tolerate a mediocre quality of synthetic speech, if they feel that a useful service is being provided. The imprecision in articulation of speech synthesized by rule is notorious, even when the suprasegmental features are determined by imitating the rhythm and intonation of a natural utterance; and of course automatic synthesis of rhythm and intonation serves only to degrade the speech quality further. Prior to the Telephone Enquiry Service, our experience of the intelligibility, gained on an informal "Listen to this: what is it saying?" basis, was quite disheartening. Rather than designing a series of formal experiments to evaluate the speech, however, we decided to introduce context in a natural way by setting up the enquiry system.

This had several effects. Firstly, the speech emanates from the telephone earpiece—the most common source of strange voices—in the familiar, comfortable surroundings of

one's own office. Compare this with the usual experience of listening to synthetic speech in a specially-prepared quiet room—or worse still, a noisy computer room—with a loudspeaker under the table as the sound source. Even the privacy of the conversation and the ability to take it at one's own pace are helpful. Secondly, there is a simple, clearly-defined universe of discourse, and the computer's response is seen in the context of the question that prompted it. As in human conversation, most of the answer to a question is already known. It is easy and natural to ask for the response to be repeated, several times if necessary, so that the idiosyncrasies of the synthetic voice can be studied in detail. Thirdly, one becomes accustomed to the speech. Acclimatization to the strange accent takes only a few minutes, but it has a very considerable and long-lasting effect on one's ability to understand it. The simple fact that synthetic speech is available on tap means that many people have heard it before. Finally, and probably most importantly, there is a strong motivation for understanding. The Telephone Enquiry Service will tell you the time, play games with you, dispense information about electronic components, even sing songs—if only you can master the art of using it. Again, how different this is from the usual situation in experiments on the perception of synthetic speech!

It is important, however, to realize the limitations of speech as an output medium. When information is needed, it is often required on paper. Reducing a person to transcribing from a computer voice to paper is not likely to result in a successful man–machine system. Furthermore, lengthy responses are inappropriate for this kind of service, as the user's attention wanders if she is not actively involved in the conversation. The following example from the "Acidosis Program", an audio response system designed to aid physicians diagnose acidosis, which uses recorded speech, represents, in our view, an inordinately long message:

> "(Chime) A VALUE OF SIX-POINT-ZERO-ZERO HAS BEEN ENTERED FOR PH. THIS VALUE IS IMPOSSIBLE. TO CONTINUE THE PROGRAM, ENTER A NEW VALUE FOR PH IN THE RANGE BETWEEN SIX-POINT-SIX AND EIGHT-POINT-ZERO (beep dah beep-beep)" (Smith & Goodwin, 1970).

The use of extraneous noises (for example, a "chime" heralds an error message, and a "beep dah beep-beep" requests data input in the form <digit> <point> <digit> <digit>) was found necessary in the Acidosis program to keep the user awake and help him with the format of the interaction. We prefer a sequential interchange of terse messages, designed to guide the caller into a state where he can rectify his error. For example:

CALLER: "6*00#".
COMPUTER: "Entry out of range."
CALLER: "6*00#" (persists).
COMPUTER: "The minimum acceptable pH value is 6·6."
CALLER: "9*03#".
COMPUTER: "The maximum acceptable pH value is 8·0."

This dialogue allows a rapid exit from the error recovery situation in the likely event that the entry has been simply mis-typed. If the error persists, the caller is given just one piece of information at a time, and forced to continue to play an active role in the interaction.

The fact that lengthy monologues are inappropriate for speech response systems means that the cardinal disadvantage of current synthesis-by-rule, namely that it is monotonous and hence difficult to listen to continuously, should rarely be encountered.

Experience with the Telephone Enquiry Service has shown that the introduction of check digits, which have been proposed for push-button entry systems (Kramer, 1970), should not be necessary with voice response. In many cases, it is easy to arrange that the result of a command will itself identify the command, enabling a keying error to be detected by the caller. In the case of data entry, audible confirmation of the entry is sufficient. The Telephone Enquiry Service includes a facility for entering numeric grades for students' assignments, and no more trouble has been experienced with erroneous inputs than in the previous hand-copying procedure. With the widespread use of electronic calculators, keying of data will become a familiar operation to most people.

Mention of calculators introduces another advantage of voice response for data entry systems. Since the feedback is auditory, one need not look up from the written data to check each input. This, combined with the full-duplex nature of the communication channel with the machine, makes data entry fast and effortless.

The Telephone Enquiry Service has some specific failings that should be mentioned. It tends to be biased towards numeric rather than alphabetic data entry. For example, each user has a "user number". There is no reason why his own name should not be employed instead. Only during the later stages of system development were users encouraged to treat their passwords as words rather than numbers. Another design problem is the restriction of access to the system. The Telephone Enquiry Service was much more popular than anticipated, especially when it was first introduced, and the one telephone line to the service was engaged for most of the working day. Users, particularly new ones, rapidly became frustrated if the service was unavailable to them. Restriction of access by rationing services, especially games, is essential to maintain fair usage of a scarce resource. A scheme of daily quotas has now been introduced. A further shortcoming of the system in its present form is the unhelpful and abrupt way most services have of announcing erroneous inputs. A simple message saying "input error" is by far the commonest response to an out-of-range number or unacceptable command. The system should detect consistent errors and generate a more constructive reply. A specific case in point is the frequent omission of "#" from the end of an input keying sequence. A simple software time-out mechanism could serve to detect this, and the caller could be politely informed of the system convention.

## 8. Conclusion

The Telephone Enquiry Service demonstrates that speech synthesis has moved from a specialist phonetic discipline into the province of engineering practicability. Despite the fairly low quality of the speech, the response from callers was most encouraging. Admittedly the user population was a self-selected sample of University staff, a body known for its tolerance to new ideas, and a system designed for the general public will require more effort to be spent on developing speech of higher intelligibility. Although it is an observed fact that some callers occasionally failed to understand parts of the responses, even after repetition, communication was largely unhindered in most cases; users being driven by a high motivation to help the system help them.

The use of speech output in conjunction with a simple input device requires careful thought for interaction to be successful and comfortable. It is necessary that the computer direct the conversation as much as possible, without seeming to be taking charge.

Provision for elimination of unwanted prompts by sophisticated users is essential to avoid frustration.

Finally, making a computer system available over the telephone results in a sudden vast increase in the user population. Although people's reaction to a new computer terminal in every office is overwhelmingly favourable, careful resource control is essential to avoid the system being hogged by a persistent few. As with all multi-access computer systems, it is particularly important that error recovery is effected automatically and gracefully.

The Telephone Enquiry Service is the fruit of efforts by many people. In particular, Brian Gaines suggested the idea initially, David Hill introduced us to speech synthesis, and Angie Corbett did a great deal of the hard work. Thanks are also due to Roger Moore, Slim Broster, and many other members of the Department of Electrical Engineering Science at Essex University for persistent field testing.

## References

ABERCROMBIE, D. (1964). Syllable quantity and enclitics in English. In ABERCROMBIE, D. *et al.*, Eds, *In Honour of Daniel Jones*, pp. 216–222. London: Longman.

BURON, R. H. (1968). Generation of a 1000-word vocabulary for a pulse-excited vocoder operating as an audio response unit. *IEEE Transactions on Audio and Electroacoustics*, **AU-26**, 21–25.

CORBETT, A. J. (1974). Telephone enquiry system using synthetic speech. *M.Sc. Thesis*. Department of Electrical Engineering Science, University of Essex, U.K.

DUDLEY, H. (1939). The vocoder. *Bell Laboratory Record*, **17**, 122–126.

GAINES, B. R. & FACEY, P. V. (1975). Some experience in interactive system development and application. *Proceedings of the IEEE*, **63**, 894–911.

HALLIDAY, M. A. K. (1967). *Intonation and Grammar in British English*. Paris: Mouton.

HALLIDAY, M. A. K. (1970). *A Course in Spoken English: Intonation*. London: Oxford University Press.

HOLMES, J. N., MATTINGLY, I. G. & SHEARME, J. N. (1964). Speech synthesis by rule. *Language and Speech*, **7**, 127–143.

HOLMES, J. N. (1973). The influence of glottal waveform on the naturalness of speech from a parallel formant synthesizer. *IEEE Transactions on Audio and Electroacoustics*, **AU-21**, 298–305.

KELLY, J. L. & GERSTMAN, L. J. (1961). An artificial talker driven from a phonetic input. *Journal of the Acoustical Society of America*, **33**, 835 (A).

KRAMER, J. J. (1970). Human factors problems in the use of pushbutton telephones for data entry. *Proceedings of Symposium on Human Factors in Telephony*, Berlin, pp. 241–258.

LAWRENCE, W. (1974). The phoneme, the syllable, and the parameter track. *Proceedings of Speech Communication Seminar*, Stockholm, August.

NEWHOUSE, A. & SIBLEY, R. A. (1969). On the use of very low cost terminals. *Proceedings of Internal Conference on Remote Data Processing*, Paris, March.

SMITH, S. L. & GOODWIN, N. C. (1970). Computer generated speech and man–computer interaction. *Journal of Human Factors and Society*, **12**, 215–223.

STOWE, A. N. & HAMPTON, D. B. (1961). Speech synthesis with pre-recorded words. *Journal of the Acoustical Society of America*, **33**, 810–811.

UMEDA, N. (1976). Linguistic rules for text-to-speech synthesis. *Proceedings of the IEEE*, **64**, 443–451.

UNDERWOOD, M. J. & MARTIN, M. J. (1976). A multi-channel formant synthesizer for computer speech response. *Proceedings of Institute of Acoustics Autumn Meeting*, Edinburgh, September.

WITTEN, I. H. (1976). Generating natural speech from text. *Proc. AISB Summer Conference*, Edinburgh, July.

WITTEN, I. H. (1977). A flexible scheme for assigning timing and pitch to synthetic speech. *Language and Speech* (to appear).

# International Journal of Man-Machine Studies

Academic Press
London New York San Francisco
A Subsidiary of Harcourt Brace Jovanovich Publishers